Computer Science Department

TECHNICAL REPORT

ITERATIVE METHODS FOR OVERFLOW
QUEUEING MODELS

By

Raymond Chan
August 1985

Technical Report #171

NEW YORK UNIVERSITY

Department of Computer Science
Courant Institute of Mathematical Sciences
251 MERCER STREET, NEW YORK, N.Y. 10012

ITERATIVE METHODS FOR OVERFLOW QUEUEING MODELS

By

Raymond Chan
August 1985

Technical Report #171

This research was supported by NSF grants NSF MCS-8203236, NSF DCR-8405506 and ONR grant N00014-83-K-0536.

Iterative Methods for Overflow Queueing Models

Author : Raymond Honfu Chan

Advisor : Olof B. Widlund

Markovian queueing networks having overflow capacity are discussed. The Kolmogorov balance equations result in a linear homogeneous system, where the right null-vector is the steady-state probability distribution for the network. The dimension of the system is equal to the total number of states in the networks which is of the same order of magnitude as the product of all the queue sizes. Thus it is not uncommon that the order of the matrix exceeds 10,000. Preconditioned conjugate gradient methods are employed to find the null-vector. The preconditioner is a *singular* matrix of the same order which can be handled by separation of variables. The resulting preconditioned system is nonsingular. Since the states remaining are precisely those at which interactions between the queues take place, the dimension of this preconditioned system can usually be reduced by an order n, where n is the individual queue size. Numerical results show that the number of iterations required for convergence is roughly constant when n increases. Analytic results are given to explain this fast convergence.

August 15, 1985.

Acknowledgments

I would like to thank Professor Olof Widlund, my thesis advisor, for his inspired guidance and his help in the preparation of this dissertation. I would also like to thank Professors Petter Bjørstad, Jonathan Goodman, Linda Kaufman, Robert Kohn and Lloyd Trefethen for their very helpful discussions.

Table of Contents

Section 1. Introduction

In Markovian queueing networks, most of the quantities of interest, for example the blocking probability and the waiting time for customers in various queues, can be expressed in terms of the steady-state probability distributions, which are the solutions of the Kolmogorov balance equations. The resulting matrix system has dimension N, where N is the total number of states in the network. This matrix has positive diagonal elements and non-positive off-diagonal elements. The graph of such a matrix derived from a q-queue model is the same as that of a q-dimensional discrete Laplacian. The matrix is non-symmetric and is known to have a one dimensional null-space. The steady-state probability distribution is the normalized right null-vector of this matrix.

However, even for systems with relatively small numbers of queues, say 4, and a small number of waiting spaces per queue, say 20, the order N of the matrix can be huge, in this case it is 16,000. Hence the matrix equations are rarely solved by direct methods such as Gaussian elimination. Kaufman [25] has considered a special direct method which handles one or serveral large submatrices by separation of variables, while the remaining variables are handled by a conventional Gaussian elimination method. Different classical iterative methods, for example, the point SOR and the block SOR methods, are also discussed there. In this paper, preconditioned conjugate gradient methods are empolyed. The preconditioner is a singular matrix of order N which can be handled by separation of variables. Although the orginal matrix is singular, we can reduce the problem to solving a non-singular inhomogeneous system by computing the component of the eigenvector which is orthogonal to the null-space of this chosen separable problem. Since the states remaining are precisely those at which interactions between the queues take place, the effective dimension of the

problem can usually be reduced by an order n, where n is the individual queue size.

In section 2, we first discuss how the balance equations are generated for different overflow queueing models, and the properties of the resulting matrices. We then discuss how to change the problems of finding the null-vectors of these matrices, which are of order $N = n^q$, into linear inhomogeneous systems of order n^{q-1}. Preconditioned conjugate gradient type iterative methods are then introduced. We discuss how to implement these iterative methods to solve the linear systems.

In section 3, we analyze the convergence rate of the method for systems with very large queue size. The convergence rate depends on the spectrum of the iteration matrices. We find that the eigenvalues of these matrices are clustered with only a few outlying eigenvalues. From these results, the fast convergence of the method follows.

In section 4, we report on the numerical results for the models discussed in section 2. A comparison is made between this method and the point SOR method. We see that our method has a much better performance. In fact, the number of iteration required to attain a given accuracy is almost constant independent of the queue size.

Section 2. The Equations and The Methods

A Markovian analysis of a queueing network based on solving the Kolmogorov equations (see Neuts [30]) for the steady-state probability distribution involves finding the null-vector of a large, sparse structured matrix. In this section, we will derive these Kolmogorov equations, construct such matrices and introduce the iterative methods used in finding the null-vectors for such matrices.

Let us first introduce the notations that we will be using. Assume that the network has q queues receiving customers from q independent Poisson sources, see Kleinrock [23]. In the i-th queue there are s_i parallel servers, and $n_i - s_i - 1$ waiting spaces. Customers enter the queue with mean arrival rate $\lambda_i > 0$. The departure distribution is independent and exponential with mean rate $\mu_i > 0$. Let $p_{i_1,i_2,\ldots,i_q}$ denote the steady-state probability distribution which gives the probability of state $(i_1,\ldots,i_q)$, i.e., the probability that i_j customers are in the j-th queue, $j=1,..,q$. Since $0 \le i_j < n_j$, $1 \le j \le q$, the total number of states in the system is $N = \prod_{j=1}^{q} n_j$.

It is important to note that, in general, there is no need to obtain the probabilities of all the states in the system. They may be used however to compute such quantities as the blocking probability, the probability of overflow from one queue to another, or the average waiting time of customers in various queues.

We begin with a simple problem, which gives the idea of how the balance equations and the corresponding matrix are generated. This problem is separable and can be solved easily. It will be used as the preconditioner for more complicated models.

§ 2.1 The Free Model

For the problems which we will discuss in this paper, the balance equations are generated by considering the rate at which a state is left and the rate and state from which that state is entered. For example, let us consider a two-queue network with no interaction between the queues. In particular, customers going into a full queue are lost. If δ_{ij} is the Kronecker delta, then the balance equations are

$$\begin{aligned}\left\{\lambda_1(1-\delta_{in_1-1}) + \lambda_2(1-\delta_{jn_2-1}) + \mu_1\min(i,s_1) + \mu_2\min(j,s_2)\right\} p_{ij} \\ = \lambda_1(1-\delta_{i0})\, p_{i-1j} + \mu_1(1-\delta_{in_1-1})\min(i+1,s_1)\, p_{i+1j} \\ + \lambda_2(1-\delta_{j0})\, p_{ij-1} + \mu_2(1-\delta_{jn_2-1})\min(j+1,s_2)\, p_{ij+1},\end{aligned} \tag{2.1.1}$$

for $0\le i<n_1$, $0\le j<n_2$. The left hand side of (2.1.1) indicates the rate at which state (i,j) is left and the right hand side indicates from which states and the rate at which state (i,j) is entered. If we write the steady-state probability distribution as

$$p_0=(p_{0,0}, p_{0,1}, \ldots, p_{0,n_2-1}, p_{1,0}, \ldots, p_{1,n_2-1}, \ldots, p_{n_1-1,n_2-1})^*,$$

where * denotes the transposition, then (2.1.1) can be written as $Dp_0 = Cp_0$ where D is diagonal and C has zero diagonal entries but non-positive off-diagonal entries. Let $A_0=D-C$. Then the steady-state probability distribution is just the right null-vector of A_0, i.e.,

$$A_0 p_0 = 0, \tag{2.1.2}$$

where A_0 is of order N, $N=n_1n_2$. Since p_0 is a probability distribution, we require

$$\sum_{j=0}^{n_2-1}\sum_{i=0}^{n_1-1} p_{ij} = 1, \tag{2.1.3}$$

$$p_{ij} \ge 0. \tag{2.1.4}$$

We will see that these constraints will uniquely determine p_0.

From (2.1.1), we see that A_0 is separable, in fact

$$A_0 = G_1 \otimes I_{n_2} + I_{n_1} \otimes G_2, \tag{2.1.5}$$

where $G_l =$

$$\begin{bmatrix} \lambda_l & -\mu_l & & & & & & \\ -\lambda_l & \lambda_l+\mu_l & -2\mu_l & & & & 0 & \\ & -\lambda_l & \lambda_l+2\mu_l & -3\mu_l & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & -\lambda_l & \lambda_l+s_l\mu_l & -s_l\mu_l & & \\ & & & & \ddots & \ddots & \ddots & \\ & 0 & & & & -\lambda_l & \lambda_l+s_l\mu_l & -s_l\mu_l \\ & & & & & & -\lambda_l & s_l\mu_l \end{bmatrix} \tag{2.1.6}$$

are matrices of order n_l, and I_k is an identity matrix of order k. Unless ambiguity arise, we will drop the subscript for I. Notice that the graph of A_0 is the same as the graph of the discrete Laplacian on a square with mesh size $(n_l-1)^{-1}$. We claim that the G_l and A_0 have one dimensional null-spaces. In fact we have (see Berman and Plemmons [5])

LEMMA 2.1.1

An irreducible matrix A with zero column sums, strictly positive diagonal and non-positive off-diagonal entries, has a one dimensional null-space. The corresponding null-vector can be chosen to have positive entries.

Proof: Let us write $A = D - C$, where D is the diagonal matrix containing all the diagonal entries of A. We have $D_{jj} > 0$ and $C_{jk} \geq 0$. Since diagonal entries do not affect the connectivity of the underlying graph of A, C and hence $C\,D^{-1}$ are irreducible. By the facts that A has column sums equal to zero and the entries of $C\,D^{-1}$ are all non-negative, $(C\,D^{-1})^*\,1 = 1$, where 1 is the vector of all ones. Hence $\| (C\,D^{-1})^* \|_\infty = 1$. Since $\| (C\,D^{-1})^* \|_\infty \geq \rho((C\,D^{-1})^*) = \rho(C\,D^{-1})$, we find that $1 \geq \rho(C\,D^{-1})$. But 1 is an eigenvalue of $(C\,D^{-1})^*$, hence $\rho(C\,D^{-1})=1$. By Perron-Frobenius theory on nonnegative irreducible matrices, see Varga [40], $\rho(C\,D^{-1}) = 1$ is a simple eigenvalue of $C\,D^{-1}$ and the corresponding eigenvector has positive entries. Hence A has a one dimensional null-space with positive null-vector. □

Clearly G_l satisfies the assumptions of the lemma. Thus it has a one dimensional null-space with positive null-vector and, by (2.1.5), so does A_0. Hence the solution p_0 to (2.1.2) - (2.1.4) exists and is unique. In particular, the positivity constraints (2.1.4) can always be satisfied. We remark that by the Gerschgorin theorem (see Varga [40]), except the zero eigenvalue, all the other eigenvalues of the G_l and hence those of A_0 are positive. In view of (2.1.5), the null-vector p_0 of A_0 can be expressed in term of the null-vector g_l of G_l as $p_0 = g_1 \otimes g_2$. To find the null-vector of G_l, we first notice that G_l can be symmetrized by a diagonal matrix. More precisely, if we define $S_l = \text{diag}\,(\,{}^l d_1, \ldots, {}^l d_{n_l}\,)$, $i=1,2$, with

$$ {}^l d_j = a_l \cdot \begin{cases} \prod_{k=1}^{j-1} \left(\dfrac{\lambda_l}{\min(k,s_l)\mu_l}\right)^{\frac{1}{2}} & 1<j\le n_l, \\ 1 & j=1, \end{cases} \tag{2.1.7} $$

then $S_l^{-1} G_l S_l$ is symmetric. Here a_l is the normalization constant such that

$$ 1_l^* S_l^2 1_l = \sum_{j=1}^{n_l} ({}^l d_j)^2 = 1, \tag{2.1.8} $$

where 1_l denotes the n_l-vector of all ones. Since G_l have zero column sums, i.e.,

$$ 1_l^* G_l = 0, \tag{2.1.9} $$

we have,

$$ G_l S_l^2 1_l = S_l^2 G_l^* 1_l = 0. \tag{2.1.10} $$

Thus $S_l^2 1_l$ is the null-vector for G_l. Hence,

$$ p_0 = S^2 1 = (S_1 \otimes S_2)^2 (1_1 \otimes 1_2). \tag{2.1.11} $$

By (2.1.8), p_0 also satisfies the summation constraint (2.1.3).

We remark that the operator A_0 has its analogy in the continuous case. It resembles the finite difference approximation to a variable coefficient elliptic operator with a transport term acting on a rectangular region with Neumann boundary conditions on every sides. A simple way to see this is to expand $p_{i,j}$ in (2.1.1) formally in Taylor's series with mesh sizes

$(n_i-1)^{-1}$ and $(n_j-1)^{-1}$. We also note that if $\lambda_i = \mu_i = s_i = 1$ in (2.1.6), then the resulting A_0 is just the usual 5-points difference operator for the Laplacian equation on a same region with Neumann boundary conditions on every sides. Thus the separability of A_0 and an analytic solution as in (2.1.11) are expected. We will return to this analogy in § 2.3.

The matrix A_0, though singular, will be used as the preconditioner for more complicated models. It is thus necessary to define an appropriate inverse of A_0. We proceed by first obtaining a spectral decomposition of A_0. Since $S_i^{-1} G_i S_i$ is symmetric, there exists orthogonal matrices Q_i and diagonal matrices Γ_i, $i=1,2$, such that

$$Q_i^* S_i^{-1} G_i S_i Q_i = \Gamma_i, \qquad i=1,2. \tag{2.1.12}$$

Here $\Gamma_i = \mathrm{diag}(\gamma_{i,1}, \gamma_{i,2}, \ldots, \gamma_{i,n_i})$ contains the eigenvalues $\{\gamma_{i,j}\}_{j=1}^{n_i}$ of G_i. By lemma 2.1.1, each G_i has only one zero eigenvalue. Let us set $\gamma_{i,n_i}=0$. By (2.1.5), A_0 can then be diagonalized by $S_1 Q_1 \otimes S_2 Q_2$. More precisely,

$$(Q_1^* S_1^{-1} \otimes Q_2^* S_2^{-1}) A_0 (S_1 Q_1 \otimes S_2 Q_2) = (\Gamma_1 \otimes I + I \otimes \Gamma_2) \equiv \Sigma, \tag{2.1.13}$$

where

$$\Sigma = \mathrm{diag}\,(\Sigma_1, \Sigma_2, \ldots, \Sigma_{n_1}),$$

with

$$\Sigma_j = \mathrm{diag}\,(\gamma_{1,j}+\gamma_{2,1}, \ldots, \gamma_{1,j}+\gamma_{2,n_2}) = \Gamma_2 + \gamma_{1,j} \cdot I, \qquad 1 \le j \le n_1. \tag{2.1.14}$$

Here Σ is diagonal and contains all the eigenvalues of A_0. Since only $\gamma_{1,n_1} = \gamma_{2,n_2} = 0$, only the last block $\Sigma_{n_1} = \Gamma_2$ is singular. From this spectral decomposition of A_0, we see that $R^N = \{p_0\} \oplus Im(A_0)$, where $Im(A_0)$ is the range of A_0.

Though Σ_{n_1} is singular, it is diagonal. Thus it is easy to define its generalized inverse, or the {1}-inverse, $\Sigma_{n_1}^+$; see Ben-Isreal and Greville [4]. In fact,

$$\Sigma_{n_1}^{-} \equiv \Gamma_2^+ = \mathrm{diag}\,(\gamma_{2,1}^{-1}, \ldots, \gamma_{2,n_2-1}^{-1}, \gamma), \tag{2.1.15}$$

with γ defined arbitrarily. Since Σ_j^{-1} is well-defined for $1 \le j < n_2$, the generalized inverse Σ^+

of Σ is given by

$$\Sigma^+ = \text{diag}\,(\Sigma_1^{-1}, \ldots, \Sigma_{n_1-1}^{-1}, \Sigma_{n_1}^{+}).$$

By (2.1.13), the generalized inverse A_0^+ of A_0 is thus given by

$$A_0^+ = (S_1 Q_1 \otimes S_2 Q_2)\, \Sigma^+ \,(Q_1^* S_1^{-1} \otimes Q_2^* S_2^{-1}). \tag{2.1.16}$$

It is important to note that A_0^+ is invertible on $Im(A_0)$. More precisely, for all $x \in Im(A_0)$, there exists a unique $y \in Im(A_0)$ such that $A_0^+ y = x$. In fact, $y = A_0 x$ and we have $y = A_0 A_0^+ y = A_0^+ A_0 y$. This follows from the fact that $y \in Im(A_0)$ if and only if the last entry of $(Q_1^* S_1^{-1} \otimes Q_2^* S_2^{-1}) y$ is zero.

The generalization to q-queue free models is immediate. By free models, we mean that there is no interaction between different queues. More precisely, customers entering a queue will be blocked and lost if that queue is full. Thus the queues are totally separated from each other. We summarize all the results in the following lemma. The proof is similar to the 2-queue case, and will not be given. For simplicity, let us denote $\bigotimes_{l=1}^{q} E_l = E_1 \otimes \cdots \otimes E_q$ and $E_l^0 = I_{n_l}$, for any arbitrary sequence of matrices $\{ E_l \}_{l=1}^{q}$, with respective dimension n_l.

LEMMA 2.1.2

The following statements hold for arbitrary q-queue free models.

(i) The generating matrix A_0 is of order $N = \prod_{l=1}^{q} n_l$, and is given by

$$A_0 = \sum_{j=1}^{q} \bigotimes_{l=1}^{q} G_l^{\delta_{lj}}, \tag{2.1.17}$$

where G_l are given by (2.1.5) and δ_{lj} is the Kronecker delta. Thus A_0 satisfies the assumptions in lemma 2.1.1.

(ii) A_0 has the following spectral decomposition

$$Q^* S^{-1} A_0 S Q = \Sigma, \tag{2.1.18}$$

where $Q = \bigotimes_{l=1}^{d} Q_l$, $S = \bigotimes_{l=1}^{d} S_l$ and $\Sigma = \sum_{j=1}^{d} \bigotimes_{l=1}^{d} \Gamma_l^{\delta_{lj}}$. Here Q_l and Γ_l are defined by (2.1.12) and S_l are defined by (2.1.7) and (2.1.8).

(iii) The steady-state probability distribution is given by

$$p_0 = S^2 \mathbf{1}, \tag{2.1.19}$$

where $\mathbf{1} = (1,1, \ldots ,1) \in R^N$.

(iv) From the spectral decomposition (ii), we have

$$R^N = \{ p_0 \} \oplus Im(A_0), \tag{2.1.20}$$

where

$$Im(A_0) = \{ x \in R^N \mid \mathbf{1}^* x = 0 \}. \tag{2.1.21}$$

(v) The generalized inverse ({1}-inverse) of A_0 is given by

$$A_0^+ = S Q \Sigma^+ Q^* S^{-1}, \tag{2.1.22}$$

where

$$(\Sigma^+)_{ii} = \begin{cases} (\Sigma_{ii})^{-1} & i<N, \\ \text{arbitrary} & i=N. \end{cases} \tag{2.1.23}$$

(vi) A_0^+ is invertible on $Im(A_0)$. More precisely, given any $x \in Im(A_0)$, there exists a unique $y \in Im(A_0)$ such that $A_0^+ y = x$ with $y = A_0 x$. We thus have

$$y = A_0 A_0^+ y = A_0^+ A_0 y \qquad \text{for all } y \in Im(A_0). \tag{2.1.24}$$

(vi) For any $p \in R^N$, there exists unique scalar α and $\xi \in Im(A_0)$, such that

$$p = \alpha p_0 + A_0^+ \xi. \ \square \tag{2.1.25}$$

Beginning in § 2.3, we will discuss models having overflow capacity. These models lead to problems that do not have analytic solutions. Thus the balance equations are usually solved by numerical methods. The iterative methods introduced in the next section will be used to solve linear systems arising from these overflow queueing networks.

§ 2.2 Preconditioned Conjugate Gradient Methods

Given a symmetric, positive definite matrix B of order m, a very attractive iterative scheme for the solution of the linear system $Bx = b$ is the conjugate gradient method. For an arbitrary initial vector x_0, the method generates a sequence of approximations $\{x_j\}$ to the solution x defined by

$$\begin{aligned} x_{j+1} &= x_j + \alpha_j d_j, \qquad & \alpha_j &= \frac{<r_j, r_j>_2}{<Bd_j, d_j>_2}, \\ r_{j+1} &= r_j - \alpha_j B d_j, \qquad & \beta_j &= \frac{<r_{j+1}, r_{j+1}>_2}{<r_j, r_j>_2}, \\ d_{j+1} &= r_{j+1} + \beta_j d_j, \end{aligned} \tag{2.2.1}$$

where $d_0 = r_0 = b - Bx_0$, and $< \cdot , \cdot >_2$ is the l_2 inner product in R^m. The method was first discussed by Hestenes and Stiefel [20], see also Luenberger [28]. The j-th iterant x_j minimizes the error functional

$$\| x - x_j \|_B = < x - x_j , B(x - x_j) >_2^{\frac{1}{2}} \tag{2.2.2}$$

over the Krylov space $x_0 + \text{Span} <r_0 , Br_0 , \ldots , B^{j-1} r_0>$. By expanding x_j, b and $x = B^{-1}b$ in the eigenvectors of B, we have

$$\frac{\|x - x_j\|_B}{\|x - x_0\|_B} \leq \min_{P \in P_{k-1}} \max_{\lambda \in \sigma(B)} | 1 - \lambda P(\lambda) | ; \tag{2.2.3}$$

see Luenberger [28]. Here P_{k-1} is the space of all polynomials of degree $k-1$ and $\sigma(B)$ is the spectrum of B. Thus in exact arithmetic, the solution can be obtained within m step. In practice, if B has a good spectrum, for example a moderate condition number or clustered eigenvalues, then the method will give a very accurate result in a few steps. An example is when B is of the form $\omega \cdot I + L + U$, where ω is a scalar, L is a matrix of low rank and U is a matrix of small norm. An approximate upper bound on the number of iterations required to make the relative error $\|x - x_j\|_B / \|x - x_0\|_B \leq \epsilon$ is

$$\left[\tfrac{1}{2} \ln(\frac{2}{\epsilon}) \right] \sqrt{\kappa(B)}, \tag{2.2.4}$$

where $\kappa(B)$ is the condition number of B. From (2.2.1), we see that this method requires $5m$ operations plus one matrix-vector multiplication per iteration. Since B is required only in the form Bd, no explicit knowledge of the matrix elements of B is required.

When the spectrum of B is not favorable, we can accelerate the rate of convergence by preconditioning the system. Given any splitting $B = B_0 + R$ of B, where B_0 is symmetric and positive definite, we can apply the conjugate gradient method to the transformed, preconditioned system $B B_0^{-1} y = b$ with $x = B_0^{-1} y$. We remark that the resulting algorithm is equivalent to applying the ordinary conjugate gradient method to the symmetric matrix $B_0^{-\frac{1}{2}} B B_0^{-\frac{1}{2}}$ but using a different basis for the given space. In fact, replacing B and every vector d in (2.2.1) by $B_0^{-\frac{1}{2}} B B_0^{-\frac{1}{2}}$ and $B_0^{\frac{1}{2}} d$, we get the same preconditioned algorithm. Notice that $B B_0^{-1} = I + R B_0^{-1}$, thus we can take advantage of the sparsity in R when we compute $R B_0^{-1} d$. We remark that we can also precondition the system from the left, i.e., we solve for x in $B_0^{-1} B x = B_0^{-1} b$. However, in each step, we have to compute $B_0^{-1} R d$ which, in general, is not sparse. The choice of B_0 depends very much on the problem itself. Since in each iteration, we have to compute a vector of the form $B_0^{-1} d$, it is necessary that the system $B_0 x = d$ can be solved economically. To speed up the convergence, B_0^{-1} should also be chosen to be an approximate inverse of B. Typically we want $B_0^{-1} B$ to be of the form $\omega \cdot I + L + U$. An analysis of this technique is given in Hestenes [21]; see also Concus, Golub and O'Leary [11]. For the generalization to positive semi-definite system, see Lewis and Rehm [26].

When B is nonsymmetric, we may observe that $B^* B$ is positive definite, symmetric and solve the normal equation $B^* B x = B^* b$ instead. Since $\kappa(B^* B) = \kappa^2(B)$, (2.2.4) suggests that if B is poorly conditioned, then the convergence may be very slow. This leads us to consider other conjugate gradient type methods. One of them is the Orthodir method discussed by Young and Jea [41]; see also Elman [15]. Given an arbitrary initial vector x_0, we generate a sequence $\{x_j\}$ of approximations to the solution x, by

$$
\begin{aligned}
x_{j+1} &= x_j + \alpha_j d_j, \qquad & \alpha_j &= \frac{<r_j, Bd_j>_2}{<Bd_j, Bd_j>_2}, \\
r_{j+1} &= r_j - \alpha_j Bd_j, & & \\
d_{j+1} &= Bd_j + \sum_{k=0}^{j} \beta_k^{(j)} d_k, \qquad & \beta_k^{(j)} &= \frac{-<B^2 d_j, Bd_k>_2}{<Bd_k, Bd_k>_2}, \qquad k \le j.
\end{aligned}
\tag{2.2.5}
$$

Here $r_0 = d_0 = b - Bx_0$. In this algorithm, we do not need to compute $B^* y$, but need extra work and storage for $\beta_k^{(j)}$ and d_k. At each step, the method chooses the x_j that minimizes the residual norm $\| B x_j - b \|_2$ over the same Krylov space $x_0 + \text{Span}<r_0, Br_0, \ldots, B^{j-1} r_0>$ that was considered previously.

The reason that we choose Orthodir method from among all the other conjugate type methods is that it is guaranted to converge even if the symmetric part of the iteration matrix is not positive-definite. There are no indications that the iteration matrices arising from queueing networks enjoy this property. From (2.2.5), the j-th iteration requires $(3(j+1)+4)m$ operations, one matrix-vector multiplication, and $(2(j+2)+2)m$ storage spaces respectively. This method is therefore practical when the number of iterations required to acquire a given accuracy remains small and roughly constant, independently of m. When this is not the case, there are two alternative ways of reducing the cost in each step, see Elman [15]. One is to restart the whole process after a certain number of iterations. The other one is to keep only the last few d_k vectors. However, neither of these algorithms is guaranteed to converge for general systems.

Similar to the symmetric case, we can split $B = B_0 + R$ where B_0 is nonsingular, not necessarily symmetric and apply the Orthodir method to the preconditioned matrices $B B_0^{-1} = I + R B_0^{-1}$ or $B_0^{-1} B = I + B_0^{-1} R$. These differ mainly in the inner products and the Krylov spaces over which we do our minimization. However, as mentioned above, if R is sparse, then the former requires much less storage and possibly less work than the latter, since for any vector x, $R B_0^{-1} x$ is sparse while $B_0^{-1} R x$ is not.

Finally let us consider the case where B is nonsymmetric and singular, see Kammerer and Nashed [22]. We note that by Fredholm's alternative, the solution to $B x = b$ will not

exist unless $b \in Im(B)$. For symmetric B, by the spectral decomposition, we know that $B|_{Im(B)}$ is invertible on $Im(B)$. This is not true for general nonsymmetric matrices. However, in § 2.6.1, we will consider a non-symmetric problem where $b \in Im(B)$ and $B|_{Im(B)}$ is invertible on $Im(B)$. We note that the equation $B|_{Im(B)}\, x = b$ can be solved by the Orthodir method without first finding $B|_{Im(B)}$ explicitly. In fact, when the initial iterant $x_0 \in Im(B)$, we have $r_0 = d_0 = b - B|_{Im(B)}\, x_0 = b - B\, x_0 \in Im(B)$. From (2.2.5), it is clear that r_j, d_j and x_j will all be in $Im(B)$ for $j > 0$. Thus there is no need to carry out the projection of B onto $Im(B)$. A suitable choice for $x_0 \in Im(B)$ is $x_0 = b$ or $x_0 = 0$.

In the following two sub-sections, we will consider two examples which will give insight in how to choose a suitable preconditioners for the queueing problems that we will discuss later.

§ 2.2.1 Preconditioning Neumann BVP by Dirichlet BVP

Let us first consider the continuous case, see Bjørstad and Widlund [7]. Let Ω be an open and bounded region in R^N with a smooth boundary $\partial\Omega$. Let η denote the outward unit normal of Ω. For any $g \in H^{\frac{1}{2}}(\partial\Omega)$, let u be the solution to

$$\begin{cases} \Delta u = 0 & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega. \end{cases} \tag{2.2.6}$$

By the regularity theorem of elliptic theory, see Lions and Magenes [27, p.189], $u \in H^1(\Omega)$. For any $w \in H^1(\Omega)$, we have, by Green's theorem,

$$\int_{\partial\Omega} \frac{\partial u}{\partial \eta}\, w = \int_{\Omega} w\, \Delta u \; + \; \int_{\Omega} \nabla u\, \nabla w = \int_{\Omega} \nabla u\, \nabla w. \tag{2.2.7}$$

Since u, $w \in H^1(\Omega)$, the right hand side is bounded. Let γ be the trace operator on $\partial\Omega$. It follows from Lions and Magenes [27, p.39] that γ is a continuous mapping from $H^1(\Omega)$ onto $H^{\frac{1}{2}}(\partial\Omega)$. In particular, $\gamma w \in H^{\frac{1}{2}}(\partial\Omega)$. Hence $\frac{\partial u}{\partial \eta}$ in (2.2.7) defines a bounded linear functional on $H^{\frac{1}{2}}(\partial\Omega)$. Thus

$$\frac{\partial u}{\partial \eta} \in H^{-1/2}(\partial\Omega), \tag{2.2.8}$$

the dual space of $H^{1/2}(\partial\Omega)$. Hence we see that this Dirichlet to Neumann map which maps g to $\frac{\partial u}{\partial \eta}$ involves a lost of a derivative in $L_2(\partial\Omega)$.

We expect that in a discrete case, such a map will have a spectral condition number proportional to the number of degrees of freedom associated with $\partial\Omega$. The following example illustrates this fact.

Consider the n by n matrices

$$U = \text{tridiag}\,(-1\,,\,2\,,\,-1\,),$$
$$V = U - e_n\, e_n^*.$$

We note that the eigenpair $\{\, Q\,,\,\Lambda\,\}$ of U is given by

$$\Lambda_j = 4\,\sin^2\left(\frac{j\pi}{2(n+1)}\right), \qquad 1 \le j \le n, \tag{2.2.9}$$

and

$$Q_{kj} = \left(\frac{2}{n+1}\right)^{1/2} \sin\left(\frac{jk\pi}{n+1}\right), \qquad 1 \le j,k \le n. \tag{2.2.10}$$

Define B and B_0 by

$$B_0 = U \otimes I_n + I_n \otimes U, \tag{2.2.11}$$
$$B = U \otimes I_n + I_n \otimes V. \tag{2.2.12}$$

We note that B_0 is the discrete Laplacian operator on $[0,1]^2$ with mesh size $h = (n+1)^{-1}$, and with Dirichlet boundary conditions on every sides. B is the same operator on the same region but with Neumann boundary condition on the side $y = 1$.

Clearly $B = B_0 - I_n \otimes e_n\, e_n^*$ and

$$(Q^* \otimes Q^*)\, B_0\, (Q \otimes Q) = (\Lambda \otimes I + I \otimes \Lambda) = \Sigma. \tag{2.2.13}$$

Let us compute the spectrum of the preconditioned matrix

$$B_0^{-1/2}\, B\, B_0^{-1/2} = I_{n^2} - B_0^{-1/2}\, (I_n \otimes e_n\, e_n^*)\, B_0^{-1/2}.$$

Using (2.2.13), we have

$$(Q^* \otimes Q^*)\, B_0^{-\frac{1}{4}}\, B\, B_0^{-\frac{1}{4}}\, (Q \otimes Q) = I - \Sigma^{-\frac{1}{4}}(I \otimes Q^*\, e_n\, e_n^*\, Q)\, \Sigma^{-\frac{1}{4}}$$
$$= \mathrm{diag}\,(B_1, \ldots, B_n), \tag{2.2.14}$$

where

$$B_j = I - \Sigma_j^{-\frac{1}{4}}\, Q^*\, e_n\, e_n^*\, Q\, \Sigma_j^{-\frac{1}{4}}, \tag{2.2.15}$$
$$\Sigma_j = \mathrm{diag}\,(\Lambda_1 + \Lambda_j, \ldots, \Lambda_n + \Lambda_j). \tag{2.2.16}$$

Notice that the second term in (2.2.15) is a rank one matrix, hence all except one of the eigenvalues of B_j are equal to 1. The outlying eigenvalue is given by

$$\lambda_1(B_j) = 1 - d_j = 1 - \|\Sigma_j^{-\frac{1}{4}}\, Q^*\, e_n\|_2^2.$$

Thus by (2.2.14), we see that all except n of the eigenvalues of $B_0^{-\frac{1}{4}}\, B\, B_0^{-\frac{1}{4}}$ are equal to 1. Those outlying eigenvalues are given by $\lambda_1(B_j)$, $1 \le j \le n$.

Using (2.2.9) and (2.2.10), we have, for $1 \le j \le n$,

$$d_j = \sum_{k=1}^{n} \frac{Q_{nk}^2}{\Lambda_k + \Lambda_j} = \left(\frac{2}{n+1}\right) \sum_{k=1}^{n} \frac{\sin^2 \frac{k\pi}{n+1}}{4\sin^2 \frac{j\pi}{2(n+1)} + 4\sin^2 \frac{k\pi}{2(n+1)}}$$
$$= \left(\frac{2}{n+1}\right) a_j \sum_{k=1}^{n} \frac{\sin^2 \frac{k\pi}{n+1}}{1 - 2a_j \cos \frac{k\pi}{n+1} + a_j^2}, \tag{2.2.17}$$

where a_j is the smallest root of

$$a_j^2 - (2 + \Lambda_j)\, a_j + 1 = 0, \qquad 1 \le j \le n.$$

Since the constant term of this quadratic equation is 1, we have,

$$a_j = 1 + \tfrac{1}{2}\Lambda_j - (\Lambda_j + \Lambda_j^2/4)^{\frac{1}{2}} = \frac{1}{1 + \tfrac{1}{2}\Lambda_j + (\Lambda_j + \Lambda_j^2/4)^{\frac{1}{2}}} < 1. \tag{2.2.18}$$

To evaluate the sum in (2.2.17), we first define, for $k \ge 0$ and $1 \le j \le n$,

$$f_j(x) = \frac{\sin^2 x}{1 - 2a_j \cos x + a_j^2},$$
$$F_k^j = \int_0^{\pi} f_j(x) \cos kx\, dx.$$

By applying the Poisson integral formula to the real harmonic function $a_j^k \cos kx$, $k \ge 0$, we have

$$\int_0^{\pi} \frac{\cos kx}{1 - 2a_j \cos x + a_j^2}\, dx = \frac{\pi a_j^k}{1 - a_j^2}.$$

Using the relation

$$\sin^2 x \cos kx = \frac{1}{2}\cos kx - \frac{1}{4}\cos (k+2)x - \frac{1}{4}\cos (m-2)x,$$

we obtain

$$F_k^j = \begin{cases} \frac{\pi}{2} & k=0, \\ \frac{1}{4}\pi a_j & k=1, \\ -\frac{1}{4}\pi a_j^{k-2}(1-a_j^2) & k>1. \end{cases} \tag{2.2.19}$$

Thus the Poisson summation formula, see Rudin [36], gives

$$\begin{aligned} d_j &= (\tfrac{2}{n+1})\, a_j\, [\, \tfrac{1}{2}f_j(0) + f_j(\tfrac{\pi}{n+1}) + f_j(\tfrac{2\pi}{n+1}) + \cdots + \tfrac{1}{2}f_j(\pi)\,] \\ &= \frac{2}{\pi}\, a_j\, [\, F_0^j + 2\sum_{k=1}^{\infty} F_{2k(n+1)}^j\,] \\ &= a_j \frac{1 - (a_j)^{2n}}{1 - (a_j)^{2n+2}}. \end{aligned} \tag{2.2.20}$$

We note that a_j in (2.2.18) decreases monotonically from 1 to $3-2\sqrt{2}$ as Λ_j varies from 0 to 4. The ratio d_j/a_j is also monotone as a function of a_j, and

$$1 - \frac{n}{n+1}a_j > \lambda_1(B_j) = 1 - d_j > 1 - a_j > 0, \qquad 1 \le j \le n. \tag{2.2.21}$$

We first claim that the n eigenvalues, $\lambda_1(B_j)$, $1 \le j \le n$, are not clustered. By (2.2.18), (2.2.9) and the inequality $\sin x \le x$ for all $x \ge 0$, we have,

$$\begin{aligned} a_j &\ge 1 - 2\sin(\tfrac{j\pi}{2(n+1)})\,(1 + \sin^2 \tfrac{j\pi}{2(n+1)})^{1/2} \\ &\ge 1 - 2\sqrt{2}\sin(\tfrac{j\pi}{2(n+1)}) \\ &\ge 1 - \frac{\sqrt{2}j\pi}{n+1}, \qquad 1 \le j \le n. \end{aligned}$$

Thus (2.2.21) gives

$$\lambda_1(B_j) \le 1 - \frac{n}{n+1}\, a_j \le \frac{\sqrt{2}(j+1)\pi}{n+1}, \qquad 1 \le j \le n. \tag{2.2.22}$$

Hence, given any $0 < \delta < 1$, the number of eigenvalues of $B_0^{-1/2} B\, B_0^{-1/2}$ lying outside the

interval [δ , 1] increases linearly with n.

Next we claim that the condition number of $B_0^{-\frac{1}{4}} B B_0^{-\frac{1}{4}}$ also increases linearly with n. By (2.2.18), (2.2.9) and the inequality $\sin \frac{j\pi}{2(n+1)} \geq \frac{j}{n+1}$ for $1 \leq j \leq n$, we have

$$a_j \leq [1 + \frac{2j^2}{(n+1)^2} + (\frac{2j}{n+1}) (1 + \frac{j^2}{(n+1)^2})^{\frac{1}{4}}]^{-1} \leq [1 + \frac{2j}{n+1}]^{-1}, \quad 1 \leq j \leq n.$$

Thus (2.2.21) gives

$$\lambda_1(B_j) \geq 1 - a_j \geq \frac{2j}{n+2j+1}, \qquad 1 \leq j \leq n.$$

Hence the condition number of $B_0^{-\frac{1}{4}} B B_0^{-\frac{1}{4}}$ is of the order $O(n)$. From (2.2.3) and (2.2.4), we see that the conjugate gradient method when applied to this preconditioned system might converge slowly for some particular initial data and right hand sides. This example shows that using a Neumann problem to precondition a Dirichlet problem leads to a non-optimal method.

§ 2.2.2 Preconditioning Oblique BVP by Neumann BVP

The queueing problems that we will discuss in the next few sections are very similar to the oblique boundary-value problems where the direction vector γ of the oblique derivative makes a constant angle with the boundary. Thus for simplicity, let us assume that

$$\gamma = \eta + \tau, \tag{2.2.23}$$

where τ is the unit tangential vector along $\partial\Omega$. Here Ω is an open bounded set in R^2, with a sufficiently smooth boundary $\partial\Omega$. In the following, let C denote any generic positive constant that depends only on Ω. Consider the following Hilbert space

$$E = \{ g \in H^{-\frac{1}{4}}(\partial\Omega) \mid \int_{\partial\Omega} g \, d\tau = 0 \}, \tag{2.2.24}$$

equipped with the usual $H^{-\frac{1}{4}}(\partial\Omega)$ norm. For any $g_1 \in E$, let u_1 be the solution to

$$\begin{cases} \Delta u = 0 & \text{in } \Omega, \\ \dfrac{\partial u}{\partial \eta} = g_1 & \text{on } \partial\Omega, \end{cases} \tag{2.2.25}$$

normalized by

$$\int_{\Omega} u = 0. \tag{2.2.26}$$

By the regularity theorem, see Lions and Magenes [27, p.189], $u_1 \in H^{\frac{1}{2}}(\Omega)$. Moreover, the following a priori bound holds:

$$\| u_1 \|_{H^{\frac{1}{2}}(\Omega)} \leq C \| g_1 \|_{H^{-\frac{1}{2}}(\partial\Omega)}. \tag{2.2.27}$$

By the trace theorem, $u_1 \in H^{\frac{1}{2}}(\partial\Omega)$ and

$$\| \frac{\partial u_1}{\partial \tau} \|_{H^{-\frac{1}{2}}(\partial\Omega)} \leq C \| u_1 \|_{H^{\frac{1}{2}}(\partial\Omega)} \leq C \| u_1 \|_{H^{\frac{1}{2}}(\Omega)}. \tag{2.2.28}$$

Thus by (2.2.23) and (2.2.27),

$$\frac{\partial u_1}{\partial \gamma} = \frac{\partial u_1}{\partial \eta} + \frac{\partial u_1}{\partial \tau} = g_1 + \frac{\partial u_1}{\partial \tau} \in H^{-\frac{1}{2}}(\partial\Omega) \tag{2.2.29}$$

and

$$\| \frac{\partial u_1}{\partial \gamma} \|_{H^{-\frac{1}{2}}(\partial\Omega)} \leq \| g_1 \|_{H^{-\frac{1}{2}}(\partial\Omega)} + \| \frac{\partial u_1}{\partial \tau} \|_{H^{-\frac{1}{2}}(\partial\Omega)} \leq C \| g_1 \|_{H^{-\frac{1}{2}}(\partial\Omega)}. \tag{2.2.30}$$

Hence the mapping T which maps g_1 to $\frac{\partial u_1}{\partial \gamma}$ is bounded. We note that $Tg_1 \in E$. In fact

$$\int_{\partial\Omega} \frac{\partial u_1}{\partial \gamma} d\tau = \int_{\partial\Omega} g_1 d\tau + \int_{\partial\Omega} \frac{\partial u_1}{\partial \tau} d\tau = \int_{\partial\Omega} \frac{\partial u_1}{\partial \tau} d\tau = \int_{\partial\Omega} du_1 = 0. \tag{2.2.31}$$

Thus T maps E into itself. We claim that T is actually a homeomorphism from E onto itself.

First we prove that T is one-to-one. This follows form the fact that the problem

$$\begin{cases} \Delta u = 0 \quad \text{in } \Omega, \\ \frac{\partial u}{\partial \gamma} = g_2 \quad \text{on } \partial\Omega, \end{cases} \tag{2.2.32}$$

with

$$\int_{\Omega} u = 0, \tag{2.2.33}$$

admits only $u_2 = 0$ as solution when $g_2 = 0$. In fact, using (2.2.32) and Green's formula

$$\int_{\Omega} |\nabla u_2|^2 = -\int_{\Omega} u_2 \Delta u_2 + \int_{\partial\Omega} u_2 \frac{\partial u_2}{\partial \eta} d\tau = -\int_{\partial\Omega} u_2 \frac{\partial u_2}{\partial \tau} d\tau = -\tfrac{1}{2}\int_{\partial\Omega} d\, u_2^2 = 0. \tag{2.2.34}$$

Thus $u_2 =$ constant. By (2.2.33), $u_2 = 0$. Hence $Tg_1 = g_2 = 0$ implies $g_1 = \frac{\partial u_2}{\partial \eta} = 0$. In

particular, T is one-to-one.

Next we prove that T is onto. We first need to determine the adjoint problem of (2.2.32). By Green's formula

$$\begin{aligned}\int_\Omega v\,\Delta u - \int_\Omega u\,\Delta v &= \int_{\partial\Omega} v\,\frac{\partial u}{\partial \eta} - \int_{\partial\Omega} u\,\frac{\partial v}{\partial \eta} \\ &= \int_{\partial\Omega} v\,\frac{\partial u}{\partial \gamma} - \int_{\partial\Omega} v\,\frac{\partial u}{\partial \tau} - \int_{\partial\Omega} u\,\frac{\partial v}{\partial \eta} \\ &= \int_{\partial\Omega} v\,\frac{\partial u}{\partial \gamma} - \int_{\partial\Omega} u\,\Big(\frac{\partial v}{\partial \eta} - \frac{\partial v}{\partial \tau}\Big),\end{aligned} \tag{2.2.35}$$

where the last equality is obtained by integration by part. Thus we see that the adjoint problem of (2.2.32) is

$$\begin{cases}\Delta u = 0 \ \text{ in } \Omega, \\ \dfrac{\partial u}{\partial \sigma} = 0 \ \text{ on } \partial\Omega,\end{cases} \tag{2.2.36}$$

where the direction vector σ is given by

$$\sigma = \eta - \tau. \tag{2.2.37}$$

Similar to (2.2.34), we see that the only solution to (2.2.36) are precisely the constant functions. Thus the compatiblity condition for problem (2.2.32) is the same as that of problem (2.2.25), namely

$$\int_{\partial\Omega} g_2\,d\tau = 0. \tag{2.2.38}$$

Hence, applying the regularity theorem again, we have, for any $g_2 \in E$, there exists a unique $u_2 \in H^1(\Omega)$, which is a solution to (2.2.32) and (2.2.33). Moreover, the following a priori bound holds:

$$\| u_2 \|_{H^1(\Omega)} \le C \| g_2 \|_{H^{-\frac{1}{2}}(\partial\Omega)}. \tag{2.2.39}$$

Since the solution u to (2.2.25) and (2.2.26) is unique, we see that

$$T\frac{\partial u_2}{\partial \eta} = g_2. \tag{2.2.40}$$

Thus T is onto.

Finally by (2.2.39) and the trace theorem, (see (2.2.28))

$$\frac{\partial u_2}{\partial \eta} = g_2 - \frac{\partial u_2}{\partial \tau} \in H^{-\frac{1}{2}}(\partial\Omega) \tag{2.2.41}$$

and

$$\| \frac{\partial u_2}{\partial \eta} \|_{H^{-\frac{1}{2}}(\partial\Omega)} \leq \| g_2 \|_{H^{-\frac{1}{2}}(\partial\Omega)} + \| \frac{\partial u_2}{\partial \tau} \|_{H^{-\frac{1}{2}}(\partial\Omega)} \leq C \| g_2 \|_{H^{-\frac{1}{2}}(\partial\Omega)}. \tag{2.2.42}$$

Hence T, the Neumann to oblique map, is a homeomorphism from E onto E. In particular, T is well-conditioned. By combining the results in this sub-section with those in § 2.2.1, it is clear that the Dirichlet to oblique map will involve a loss of a derivative in $L_2(\partial\Omega)$. Hence we see that the Neumann problems are better suited as a preconditioner for the oblique derivative problems than the Dirichlet problems.

In § 3.9, we will consider a discrete version of preconditioning an oblique BVP by a Neumann BVP in a rectangular region. We will see that the singular values of the preconditioned matrix are clustered around $\sqrt{2}$. Using (2.2.3), we are then able to prove the optimality of the method.

§ 2.3 Two-Queue Overflow Models

In this section, we will discuss 2-queue networks with built-in overflow capacity. We assume here that overflow from a queue are permitted only when that queue is full. In § 2.6, we will discuss models in which overflow can occur even before the queue is full. Thus here we consider only two types of overflow models. In one we allow overflow in only one direction, in the other in both. The method introduced here will be generalized to networks with more than two queues in § 2.5.2.

§ 2.3.1 Overflow In One Direction

Let us consider the model in which overflow is permitted only from the first queue into the second. More precisely, customers entering the first queue will wait and be served by the second queue if all the spaces in the first queue are occupied. On the other hand, customers entering the second queue are lost if the second queue is full. This model is discussed in Kaufman [25]. The balance equations for this model are given by:

$$\begin{aligned}\left\{\lambda_1(1-\delta_{in_1-1}\delta_{jn_2-1}) + \lambda_2(1-\delta_{jn_2-1}) + \mu_1 \min(i,s_1) + \mu_2 \min(j,s_2)\right\} p_{ij} \\ = \lambda_1(1-\delta_{i0})\, p_{i-1j} + \mu_1(1-\delta_{in_1-1}) \min(i+1,s_1)\, p_{i+1j} \\ + (\lambda_1\delta_{in_1-1} + \lambda_2)(1-\delta_{j0})\, p_{ij-1} + \mu_2(1-\delta_{jn_2-1}) \min(j+1,s_2)\, p_{ij+1},\end{aligned} \tag{2.3.1}$$

for $0 \le i < n_1$, $0 \le j < n_2$. This differs from (2.1.1) only in the coefficients of p_{ij} and p_{ij-1}. The coefficient of p_{ij-1} indicates that customers are gained in the second queue at a rate λ_2, but if the first queue is full, additional customers will also arrive at the second queue at the rate λ_1.

Let $p = (p_{0,0}, \ldots, p_{n_1-1,n_2-1})^*$ be the steady-state probability distribution vector for this problem. Using the notations in § 2.1, we are solving a homogeneous system of order

$N = n_1 n_2$, namely,

$$A\,p = (A_0 + R_0)\,p = 0, \tag{2.3.2}$$

$$\sum_{l=1}^{n_1 - 1} \sum_{j=0}^{n_2 - 1} p_{lj} = 1, \tag{2.3.3}$$

$$p_{lj} \geq 0. \tag{2.3.4}$$

Here

$$R_0 = ({}^1e_{n_1}\,{}^1e_{n_1}^*) \otimes {}^2R_1, \tag{2.3.5}$$

with

$${}^jR_l = \lambda_l \cdot \begin{bmatrix} 1 & & & & \\ -1 & 1 & & 0 & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & 0 & & -1 & 1 \\ & & & & -1\ \ 0 \end{bmatrix}, \tag{2.3.6}$$

a square matrix of order n_j. ${}^1e_j$ denotes the j-th unit vector in R^{n_1}.

By (2.3.5) and the fact that A_0 satisfies the assumptions of lemma 2.1.1, A also satisfies these assumptions. Thus A has a one dimensional null-space with a positive null-vector. Hence the solution p to (2.3.2) - (2.3.4) exists and is unique. We remark that A and A_0 have the same graph. Since A_0 and A both have zero column sums and one dimensional null-spaces, it follows that

$$Im(A) = Im(A_0) = \{x \in R^N \,|\, 1^*\,x = 0\}, \tag{2.3.7}$$

where $1 = (1,1,\ldots,1)^* \in R^N$. Since $R_0 = A - A_0$, $Im(R_0) \subset Im(A_0)$. By (2.3.5), it is clear that $Im(R_0)$ is the set of all $x \in Im(A_0)$ with the first $(n_1-1)n_2$ entries equal to zero. More precisely, by (2.3.7),

$$Im(R_0) = \{x \in R^N \mid x = {}^1e_{n_1} \otimes y \text{ where } y \in R^{n_2} \text{ with } \textstyle\sum y_j = 0\}. \tag{2.3.8}$$

By (2.1.25), there exists a unique α and $\xi_0 \in Im(A_0)$ such that $p = \alpha\,p_0 + A_0^+\,\xi_0$. Since $A_0^+\,\xi_0 \in Im(A_0)$, we have, by (2.3.7), $1^*\,p = 1^*\,\alpha\,p_0 + 1^*\,A_0^+\,\xi_0 = \alpha\,1^*\,p_0$. By the

summation constraints (2.1.3) and (2.3.3), $1^* p_0 = 1^* p = 1$. Thus $\alpha = 1$ and

$$p = p_0 + A_0^+ \xi_0. \tag{2.3.9}$$

Substituting (2.3.9) into (2.3.2) and using (2.1.24), we have

$$(I + R_0 A_0^+) \xi_0 = -R_0 p_0. \tag{2.3.10}$$

By (2.3.5) and (2.1.11), it is easily checked that $R_0 p_0 \neq 0$. Thus the problem of finding a null-vector to (2.3.2) has been transformed into the problem of solving a linear inhomogeneous system (2.3.10).

We remark that $(I + R_0 A_0^+)\xi = A A_0^+ \xi$ for all $\xi \in Im(A_0)$. Thus we are preconditioning the equation (2.3.2) by A_0 from the right. Notice that the right hand side of (2.3.10) is the product of R_0 and the null-vector of the preconditioner A_0. Thus it is impossible to precondition the matrix A by a non-singular preconditioner. If that were done, the right hand side of the preconditioned system would be zero and we would be solving a homogeneous problem rather than an inhomogeneous one. However, after fixing one degree of freedom in the solution p, it is always possible to design a non-singular preconditioner for the resulting reduced system. This follows from the fact that the matrix A has a null-space of dimension one, hence the resulting reduced system will no longer be singular. However, numerical results show that such non-singular preconditioners are in general not as good as the singular one. We will return to this topic at the end of this section and in § 2.4. Let us remark that, although $A A_0^+$ is singular, the matrix $(I + R_0 A_0^+)$ is nonsingular.

LEMMA 2.3.1

Consider a system of the form

$$\begin{cases} A p = 0 \\ 1^* p = 1 \\ p_j \geq 0, \end{cases}$$

where A is of the same dimension as A_0. If the solution p exists and is unique and $1^* A = 0$ then the matrix $(I + R_0 A_0^+)$ is non-singular, where $R_0 = A - A_0$.

Proof: We first note that $\mathbf{1}^* A = \mathbf{0}$ implies $Im(R_0) \subseteq Im(A_0)$. Hence $(I + R_0 A_0^+)$ maps $Im(A_0)$ into itself. Moreover, the existence and uniqueness of p implies the existence and uniqueness of a $\xi_0 \in Im(A_0)$ that satisfies

$$(I + R_0 A_0^+)\xi_0 = -R_0 p_0.$$

Thus the matrix is invertible in $Im(A_0)$. Suppose y is in the kernel of this matrix. By (2.1.20), there exists a unique β and $x \in Im(A_0)$ such that $y = \beta p_0 + x$. Hence $(I + R_0 A_0^+) y = 0$ implies that $-\beta p_0 = (I + R_0 A_0^+) x + \beta R_0 A_0^+ p_0$. Since $Im(R_0) \subseteq Im(A_0)$, the right hand side is in $Im(A_0)$. Thus by (2.1.20), $\beta = 0$ and $(I + R_0 A_0^+) x = 0$. Since $x \in Im(A_0)$, the last equation implies $x = 0$. Hence $y = 0$. Thus the matrix is non-singular. □

In essence, the singularity of A has been cancelled by the singularity of A_0. We remark that if A is a generating matrix, then by the conservation of flow in and out of any given state, the assumption $\mathbf{1}^* A = \mathbf{0}$ always holds, see Messay [29].

By this lemma, it is legitimate to solve the inhomogeneous system (2.3.10). By (2.3.10), $\xi_0 = R_0(p_0 - A_0^+ \xi_0) \in \mathrm{Im}(R_0)$. By (2.3.8), $\xi_0 = {}^1e_{n_1} \otimes y_0$ where $y_0 \in R^{n_2}$ and there are only n_2 degrees of freedom in ξ_0. This suggests that the system (2.3.10), which is of order $N = n_1 n_2$, can be reduced to a system of order n_2. To achieve this, we first denote the projection from $\mathrm{Im}(R_0)$ onto R^{n_2} by E^*. By (2.3.5),

$$E = {}^1e_{n_1} \otimes I_{n_2}. \tag{2.3.11}$$

We note that $E^* \xi_0 = y_0$ and $E E^* \xi_0 = E y_0 = \xi_0$. Premultiplying (2.3.10) by E^*, we get,

$$E^*(I + R_0 A_0^+) E E^* \xi_0 = E^* R_0 p_0.$$

This is an n_2 by n_2 system of the form $B y_0 = b$, where

$$B = E^*(I + R_0 A_0^+) E = I_{n_2} + {}^2R_1 E^* A_0^+ E, \tag{2.3.12}$$

$$b = E^* R_0 p_0 = {}^2R_1 E^* p_0 = ({}^1d_{n_1})^2 \cdot {}^2R_1 S_2^2 \mathbf{1}_2. \tag{2.3.13}$$

Here ${}^1d_{n_1}$ is given by (2.1.7). By (2.3.9) and (2.1.11), the original null-vector p is given by

$$p = p_0 + A_0^+ E y_0 = (S_1 \otimes S_2)^2 1 + A_0^+ E y. \tag{2.3.14}$$

Since B is non-symmetric, we can solve $B y_0 = b$ by using the normal equations $B^* B y_0 = B^* b$, or by the Orthodir method discussed in § 2.2.

For A_0 to be a good preconditioner, it is necessary that $A_0^+ x$ can be easily computed for all $x \in Im(A_0)$. In view of (2.3.12), we in fact only require that $E^* A_0^+ E y$ can be computed economically for all $y \in E^*(Im(A_0))$. There are three alternative ways of computing this quantity. For clarity, we introduce only one alternative here, and leave the other two to Appendix A.1. Let us remark that the alternative discussed here is the easiest to handle analytically, but that it is not the best choice for numerical work.

Alternative (A):- Complete diagonalization of A_0 *by* $(S_1 Q_1 \otimes S_2 Q_2)$.

Using (2.1.16) and some straightforward computations, we have

$$\begin{aligned} E^* A_0^+ E &= E^* (I \otimes S_2 Q_2) (S_1 Q_1 \otimes I) \Sigma^+ (Q_1^* S_1^{-1} \otimes I) (I \otimes Q_2^* S_2^{-1}) E \\ &= S_2 Q_2 E^* (S_1 Q_1 \otimes I) \Sigma^+ (Q_1^* S_1^{-1} \otimes I) E Q_2^* S_2^{-1} \\ &= S_2 Q_2 \Phi Q_2^* S_2^{-1}, \end{aligned} \tag{2.3.15}$$

where Φ is diagonal and is given by

$$\Phi = \sum_{j=1}^{n_1-1} ({}^1q_{n_1,j})^2 \cdot \Sigma_j^{-1} + ({}^1q_{n_1,n_1})^2 \cdot \Sigma_{n_1}^+. \tag{2.3.16}$$

Here ${}^1q_{n_1,j}$ denotes the (n_1,j) entry of the orthogonal matrix Q_1. Σ_j and $\Sigma_{n_1}^+$ are given by (2.1.14) and (2.1.15). Putting (2.3.15) into (2.3.12), we have

$$B = I + {}^2R_1 S_2 Q_2 \Phi Q_2^* S_2^{-1}. \tag{2.3.17}$$

Thus, before we start the iteration, we have to generate $\{Q_l,\Gamma_l\}_{l=1}^2$, the eigenpairs of the symmetric matrices $S_l^{-1} G_l S_l$. This may be done by calling a standard eigenvector subroutine, see the EISPACK manual by Smith [38]. Since the G_l are tridiagonal and the S_l are diagonal, this requires $O(n_2^2)$ operations. Since Σ_j^{-1}, $1 \le j < n_1$, and $\Sigma_{n_1}^+$ are diagonal, Φ can be generated in $O(n_2^2)$ operations from (2.3.16). With Φ and Q_2 generated and stored before we

iterate, the computation of $(E^* A_0^+ E)\, y$ requires only $2\, n_2^2 + n_2$ operations for any vector y. The storage requirement is $n_1^2 + n_2^2 + O(n_1)$, since we need to store the Q_l's. By (2.3.17), as ${}^2R_1$ is a bidiagonal matrix, the matrix-vector multiplication By thus requires approximately $2\, n_2^2 + O(n_2)$ operations. Notice that there is no need to compute the last entry of Φ. In fact,

LEMMA 2.3.2

Φ_{n_2}, the last diagonal entry of Φ, can be defined arbitrarily.

Proof: Since $\gamma_{2,n_2} = 0$, it follows from (2.3.16), (2.1.14) and (2.1.15) that

$$\Phi_{n_2} = \sum_{j=1}^{n_1-1} ({}^1q_{n_1,j})^2 (\gamma_{1,j})^{-1} + ({}^1q_{n_1,n_1})^2\, \gamma, \tag{2.3.18}$$

where γ is arbitrary. The lemma is therefore true if ${}^1q_{n_1,n_1} \neq 0$. Since ${}^1q_{\cdot,n_1}$ is the eigenvector corresponding to the zero eigenvalue of $S_1^{-1} G_1 S_1$, thus by (2.1.10), ${}^1q_{\cdot,n_1} = S_1\, 1_1$. In particular, by (2.1.7), ${}^1q_{n_1,n_1} = a_1 \cdot \prod_{k=1}^{n_1-1} \left(\frac{\lambda_1}{\min(k,s_1)\ \mu_1}\right)^{\frac{1}{2}} \neq 0$. □

Combining this result and the fact that $Im(R_0)$ is an (n_2-1)-dimensional vector space, we can further reduce the dimension of the system (2.3.10) to (n_2-1). We will exploit this fact in § 3.

When n_2 is of moderate size, (2.3.17) suggests we can compute and store B, and then solve $B\, y_0 = b$ by a direct method such as Gaussian elimination. However, the numerical results given in § 4 are computed by using iterative methods because we are also interested in the case when n_2 is very large. Since each matrix-vector multiplication Bd requires about $2\, n_2^2$ operations, thus the work required for the Orthodir method in the j-th iteration is roughly $3\, j\, n_2 + 2\, n_2^2$, see § 2.2. The algorithm converges very fast, and the number of iterations required for a given accuracy is roughly independent of n_2. Solving the normal equations would require $4n_2^2 + O(n_2)$ operations per iteration. However, the number of iterations is about half of that required by the Orthodir method.

After obtaining the solution to $B\, y_0 = b$ by an iterative method, we can generate the original null-vector p by (2.3.14). We note that this step may require $N\ (n_1 + n_2)$ operations since $A_0^+\, E\, y_0$ is not sparse and N storage spaces are required for holding p. However, in some particular but interesting cases, we can reduce both the operation count and storage. For example, if only the blocking probability p_{n_1-1,n_2-1} is required, then we only need to calculate an expression of the form $e_N^*\, A_0^+\, E\, y$, where e_N is the N-th unit vector in R^N. This requires even fewer components of the solution than $E^*\, A_0^+\, E\, y$ and can be evaluated in $O(n_2^2)$ operations and requires only $O(n_2)$ storage. As another example, suppose we want the probability of overflow from queue 1 to queue 2, which is given by $\sum_{j=0}^{n_2-2} p_{n_2-1,j}$, or more generally, suppose only a linear functional of p is required. In these cases, there is no need to generate and store p explicitly. The idea is to generate the solution one block at a time, and then accumulate its contribution to the functional before we generate another block. More precisely, suppose we want to calculate $l^*\, p$, where l is a vector in R^N. By (2.3.14) and the fact that the entries of S_l are given by (2.1.7), we only need to evaluate an expression of the form

$$s = l^*\, (A_0^+\, E\, y_0) = l^*\, (S_1\, Q_1 \otimes S_2\, Q_2)\, \Sigma^+\, (Q_1^*\, S_1^{-1} \otimes Q_2^*\, S_2^{-1})\, E\, y_0.$$

Let us first partition l into n_1 block, $\hat{l}_1, \ldots, \hat{l}_{n_1}$, each with n_2 entries, and define

$$w_j = \begin{cases} {}^1q_{n_1 j}\, ({}^1d_{n_1})^{-1} \cdot S_2\, Q_2\, \Sigma_j^{-1}\, (Q_2^*\, S_2^{-1}\, y_0), & 1 \le j < n_1, \\ {}^1q_{n_1 n_1}\, ({}^1d_{n_1})^{-1} \cdot S_2\, Q_2\, \Sigma_{n_1}^{+}\, (Q_2^*\, S_2^{-1}\, y_0), & j = n_1, \end{cases} \tag{2.3.19}$$

then it is easy to check that

$$s = \sum_{j=1}^{n_1} \sum_{k=1}^{n_1} ({}^1d_k\, {}^1q_{kj})\, \hat{l}_k^*\, w_j.$$

Thus it is clear that we can generate w_j one at a time and accumulate its contribution to $l^*\, p$ before we generate another w_j. Hence no extra storage is required for p. Notice also that this way of accumulating the result blockwise does not increase the work. Similar techniques

are also discussed in Banegas [3].

Let us consider an important special case, the single server case, where $s_l = 1$. In this case, we can derive explicit formula for Q_l, Γ_l and Φ. We will show that $Q_l x$ can be computed by using the Fast Fourier Transform. Hence the operations count and storage required for each iteration can be further reduced by almost a factor of n_l.

We first give the formula for Q_l and Γ_l, $l = 1, 2$. By (2.1.6), with $s_l = 1$, we have

$$S_l^{-1} G_l S_l = \sqrt{\lambda_l \mu_l}\cdot \begin{bmatrix} \rho_l & -1 & & & \\ -1 & \rho_l+\frac{1}{\rho_l} & -1 & & 0 \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & 0 & & -1 & \rho_l+\frac{1}{\rho_l} \quad -1 \\ & & & & -1 \quad \frac{1}{\rho_l} \end{bmatrix} \tag{2.3.20}$$

for $l=1,2$, where $\rho_l = (\frac{\lambda_l}{\mu_l})^{1/2}$. Let $\theta_{l,j} = \frac{j\pi}{n_l}$, and define $\psi_{l,j}$ by

$$\sin(\psi_{l,j}-\theta_{l,j}) = \frac{1}{\rho_l}\sin\psi_{l,j}, \tag{2.3.21}$$

for $1\le j<n_l$, or equivalently,

$$\sin^2\psi_{l,j} = \frac{\rho_l^2 \sin^2\theta_{l,j}}{1-2\rho_l\cos\theta_{l,j}+\rho_l^2}. \tag{2.3.22}$$

It is easy to check that

$${}^l q_j = \begin{cases} (\frac{2}{n_l})^{1/2}(\sin\psi_{l,j}\ ,\ \sin(\theta_{l,j}+\psi_{l,j})\ ,\ \ldots\ ,\ \sin((n_l-1)\theta_{l,j}+\psi_{l,j}))^* & 1\le j<n_l, \\ (\frac{1-\rho_l^2}{1-\rho_l^{2n_l}})^{1/2}(1\ ,\ \rho_l\ ,\ \ldots\ ,\ \rho_l^{n_l-1})^* & j=n_l, \end{cases} \tag{2.3.23}$$

are the normalized eigenvectors of $S_l^{-1} G_l S_l$ with eigenvalues

$$\gamma_{l,j} = \begin{cases} \sqrt{\lambda_l\mu_l}(\rho_l+\frac{1}{\rho_l}-2\cos\theta_{l,j}) & 1\le j<n_l, \\ 0 & j=n_l. \end{cases} \tag{2.3.24}$$

Thus $Q_l = ({}^lq_1, {}^lq_2, \ldots, {}^lq_{n_l})$ can be generated without calling any EISPACK subroutine.

Next we claim that, for any real vector x, $Q_l x$, $l=1,2$ can be computed by using the Fast Fourier Transform. In fact, the k-entry of this vector is given by

$$\begin{aligned}(Q_l x)_k &= \sum_{j=1}^{n_l} {}^lq_{kj}\, x_j \\ &= (\frac{2}{n_l})^{1/4} \sum_{j=1}^{n_l-1} \sin((k-1)\theta_{lj}+\psi_{lj})\, x_j + (\frac{1-\rho_l^2}{1-\rho_l^{2n_l}})^{1/4} \rho_l^k\, x_{n_l} \\ &= (\frac{2}{n_l})^{1/4}\ \mathrm{IMAG}\left\{\sum_{j=1}^{n_l-1} e^{l(k-1)\theta_{lj}}\cdot(e^{l\psi_{lj}} x_j)\right\} + (\frac{1-\rho_l^2}{1-\rho_l^{2n_l}})^{1/4} \rho_l^k\, x_{n_l},\end{aligned}$$

where IMAG means taking the imaginary part. The numbers $z_j = e^{l\psi_{lj}} x_j$, $j=1,\ldots,n_l$ can be computed by using n_l complex multiplications. The expression $\sum_{j=1}^{n_l} e^{l(k-1)\theta_{lj}} z_j$ can be evaluated by the Fast Fourier Transform. This requires only $O(n_l \log n_l)$ operations for arbitrary n_l, see Appendix A.4. We remark that when using the Fast Fourier Transform, there is no need to store Q_l. Thus the storage requirement is reduced from $O(n_l^2)$ to $O(n_l)$. From (2.3.17), we see that the work and storage required for computing the matrix-vector multiplication Bd are thus reduced to $O(n_2 \log n_2)$ and $O(n_2)$ respectively. The work of generating the whole null-vector p can also be reduced to $O(n_2^2 \log n_2)$. If only one of the $s_l = 1$, we can still reduce the work and storage by using alternatives B or C given in Appendix A.1.

Finally we give a formula for Φ. First we recall from lemma 2.3.2 that Φ_{n_2} can be set arbitrarily. For $1 \le j < n_2$, by (2.3.16), the j-th diagonal entry of Φ is given by

$$\Phi_j = \sum_{k=1}^{n_1} ({}^1q_{n_1,k})^2\, (\gamma_{1,k} + \gamma_{2,j})^{-1}.$$

By using the formulas for Q_2, γ_{lj} and ψ_{lj} in (2.3.22), (2.3.23), and (2.3.24), we get

$$\Phi_j = \frac{2}{n_1} \sum_{k=1}^{n_1-1} \frac{\sin^2((n_1-1)\,\theta_{1,k} + \psi_{1,k})}{(\gamma_{1,k} + \gamma_{2,j})} + \frac{1-\rho_1^2}{1-\rho_1^{2n_1}} \cdot \frac{\rho_1^{2n_1-2}}{\gamma_{2,j}}$$

$$= \frac{2}{n_1}\sum_{k=1}^{n_1-1}\frac{\sin^2(\psi_{1,k}-\theta_{1,k})}{(\gamma_{1,k}+\gamma_{2,J})} + \frac{1-\rho_1^2}{1-\rho_1^{2n_1}}\cdot\frac{\rho_1^{2n_1-2}}{\gamma_{2,J}}$$

$$= \frac{2}{n_1}\frac{1}{\rho_1^2}\sum_{k=1}^{n_1-1}\frac{\sin^2\psi_{1,k}}{(\gamma_{1,k}+\gamma_{2,J})} + \frac{1-\rho_1^2}{1-\rho_1^{2n_1}}\cdot\frac{\rho_1^{2n_1-2}}{\gamma_{2,J}}$$

$$= \frac{2}{n_1}\sum_{k=1}^{n_1-1}\frac{\sin^2\theta_{1,k}}{(1-2\rho_1\cos\theta_{1,k}+\rho_1^2)\,(\mu_1(1-2\rho_1\cos\theta_{1,k}+\rho_1^2)\;+\gamma_{2,J})} + \frac{1-\rho_1^2}{1-\rho_1^{2n_1}}\cdot\frac{\rho_1^{2n_1-2}}{\gamma_{2,J}}$$

$$= \frac{2}{n_1}\frac{1}{\gamma_{2,J}}\sum_{k=1}^{n_1-1}\frac{\sin^2\theta_{1,k}}{(1-2\rho_1\cos\theta_{1,k}+\rho_1^2)} - \frac{2}{n_1}\frac{\mu_1}{\gamma_{2,J}}\sum_{k=1}^{n_1-1}\frac{\sin^2\theta_{1,k}}{\mu_1(1-2\rho_1\cos\theta_{1,k}+\rho_1^2)\;+\gamma_{2,J}}$$

$$+ \frac{1-\rho_1^2}{1-\rho_1^{2n_1}}\cdot\frac{\rho_1^{2n_1-2}}{\gamma_{2,J}}.$$

Using the Poisson summation formula (see (2.2.17) and (2.2.20)), the first term is equal to $\frac{(1-\rho_1^{2n_1-2})}{\gamma_{2,J}(1-\rho_1^{2n_1})}$, which combines with the third term and gives

$$\Phi_J = \frac{1}{\gamma_{2,J}} - \frac{2}{n_1}\frac{\mu_1}{\gamma_{2,J}}\sum_{k=1}^{n_1-1}\frac{\sin^2\theta_{1,k}}{\mu_1(1-2\rho_1\cos\theta_{1,k}+\rho_1^2)\;+\gamma_{2,J}}. \tag{2.3.25}$$

The second term in (2.3.25) can also be computed by using the Poisson summation formula. To apply the formula, we notice that

$$(1-2\rho_1\cos\theta_{1,k}+\rho_1^2)\;+\frac{\gamma_{2,J}}{\mu_1} = \rho_1\left\{\rho_1+\frac{1}{\rho_1}-2+\frac{\gamma_{2,J}}{\sqrt{\lambda_1\mu_1}}+4\sin^2\frac{\theta_{1,k}}{2}\right\}$$

$$= \frac{\rho_1}{a_J}\left\{1-2a_J\cos^2\theta_{1,k}\;+a_J^2\right\},$$

where a_J is the smallest root of

$$a_J^2-(\rho_1+\frac{1}{\rho_1}+\frac{\gamma_{2,J}}{\sqrt{\lambda_1\mu_1}})a_J+1=0. \tag{2.3.26}$$

Thus,

$$\frac{2}{n_1}\sum_{k=1}^{n_1-1}\frac{\sin^2\theta_{1,k}}{(1-2\rho_1\cos\theta_{1,k}+\rho_1^2)+\frac{\gamma_{2,J}}{\mu_1}} = \frac{2}{n_1}\frac{a_J}{\rho_1}\sum_{k=1}^{n_1-1}\frac{\sin^2\theta_{1,k}}{1-2a_J\cos\theta_{1,k}+a_J^2}$$

$$= \frac{a_j}{\rho_1}\left(\frac{1-a_j^{2n_1-2}}{1-a_j^{2n_1}}\right).$$

Put this back into (2.3.25) and we have

LEMMA 2.3.3

In the single server case,

$$\Phi_j = \begin{cases} \dfrac{1}{\gamma_{2j}}\left\{1 - \dfrac{a_j}{\rho_1}\left(\dfrac{1 - a_j^{2n_1-2}}{1 - a_j^{2n_1}}\right)\right\} & 1 \le j < n_2, \\ \text{arbitrary} & j = n_2, \end{cases} \tag{2.3.27}$$

where a_j are given by (2.3.25). □

Thus Φ can be generated in $O(n_2)$ operations.

Before we turn to other problems, let us remark that the matrix A has an analogy in the continuous case. Notice that ${}^2R_1$ in (2.3.6) is the forward difference operator on a line. Thus R_0 in (2.3.5) resembles an operator which is zero in a rectangular region, but with a tangential derivative along one of the side. This particular side corresponds to those states where the first queue is full. Recall that A_0 resembles a finite difference approximation of an elliptic operator with a transport term on the same region and with Neumann boundary conditions on every sides. Thus the continuous analogy of $A = A_0 + R_0$ is the finite difference approximation of the same operator as A_0, but with an oblique derivative on the particular side. Using this analogy and the result obtained in § 2.2.1 and § 2.2.2, an intuitive explanation for the fast convergence of the method is that the preconditioner A_0 is a good approximation to the operator A, in the sense that it changes its oblique boundary condition into a Neumann boundary condition. In § 2.4, we will design preconditioners that change the oblique boundary condition to a Dirichlet or a mixed type boundary condition. Numerical

results in § 4 show that the method converges faster if the preconditioner has the same or similar type of boundary condition as the original operator.

In the single server case, the underlying elliptic operator has constant coefficients and is given by

$$(\lambda_1 + \mu_1)\, p_{xx} + (\lambda_2 + \mu_2)\, p_{yy} + 2(n_1-1)(\mu_1 - \lambda_1)\, p_x + 2(n_2-1)(\mu_2 - \lambda_2)\, p_y \approx 0. \tag{2.3.28}$$

Thus for n_l large, it is sensible to consider a limit such that $\mu_l - \lambda_l = O((n_l-1)^{\alpha})$ for a certain α. In § 3, we will analyse the method under this limit. In the next sub-section, we will consider models where overflow is permitted in both directions.

§ 2.3.2 Overflow In Both Directions

A more interesting model is to permit overflow in both directions. More precisely, we assume that whenever a queue is full, customers entering that queue will be served by the other queue. When both queues are full, customers entering the network are lost.

Let $\bar{p}$ denote the steady-state probability distribution for this model. Then the balance equations, in matrix term, can be written as,

$$\bar{A}\,\bar{p} = (A_0 + \bar{R})\,\bar{p} = 0, \tag{2.3.29}$$

with constraints

$$1^{*}\,\bar{p} = 1, \tag{2.3.30}$$

$$\bar{p}_{I,J} \geq 0. \tag{2.3.31}$$

Here,

$$\bar{R} = ({}^{1}e_{n_1}\,{}^{1}e_{n_1}^{*}) \otimes {}^{2}R_1 + {}^{1}R_2 \otimes ({}^{2}e_{n_2}\,{}^{2}e_{n_2}^{*}), \tag{2.3.32}$$

where ${}^{J}R_l$ are given by (2.3.6). This system differs from the one we have in § 2.3.1 only in $\bar{R}$. The second term in $\bar{R}$ corresponds to the overflow from queue 2 to queue 1. Notice that $\bar{R}$ is still sparse and $\bar{A}$ still satisfies the assumption of lemma 2.1.1. Thus $\bar{A}$ has a one

dimensional null-space with positive null-vector. Hence the solution $\bar{p}$ exists and is unique. We note that

$$Im(\bar{R}) = \{ x \in R^N \mid x = {}^1e_{n_1} \otimes {}^2y + {}^1y \otimes {}^2e_{n_2},\ {}^iy \in R^{n_i},\ \sum_{ij} ({}^iy)_j = 0 \} \subset Im(A_0). \qquad (2.3.33)$$

Moreover, by (2.1.25), (2.1.21) and (2.3.30) there exists a unique $\bar{\xi}_0 \in Im(A_0)$ such that

$$\bar{p} = p_0 + A_0^+ \bar{\xi}_0. \qquad (2.3.34)$$

Substituting this into (2.3.29) and using (2.1.24), we get

$$(I + \bar{R} A_0^+) \bar{\xi}_0 = -\bar{R} p_0. \qquad (2.3.35)$$

It is easy to check that $\bar{R} p_0 \neq 0$. Thus the homogeneous system (2.3.29) is again transformed into an inhomogeneous one. By lemma 2.3.1, we see that the matrix $(I + \bar{R} A_0^+)$ is nonsingular. By (2.3.35), $\bar{\xi}_0 = -\bar{R} \bar{p} \in Im(\bar{R})$. Thus $\bar{\xi}_0$ has at most $m = n_1 + n_2$ degrees of freedom. Let us denote by E^* the projection from $Im(\bar{R})$ onto R^m, and let $\bar{y}_0 = E^* \bar{\xi}_0$. We have $E E^* \bar{\xi}_0 = E \bar{y}_0 = \bar{\xi}_0$. Notice that for any $\xi \in Im(\bar{R})$ and $y \in R^m$, the computation of $E^* \xi$ and $E y$ requires no arithmetic operations. In fact, it can be done by deleting or inserting zeros in the vector ξ and y respectively. Premultiplying (2.3.35) by E^*, we obtain an m by m system

$$(I + E^* \bar{R} A_0^+ E) \bar{y}_0 = -E^* \bar{R} p_0, \qquad (2.3.36)$$

which can be solved by the iterative methods previously considered. After obtaining $\bar{y}_0$, the solution $\bar{p}$ is given by

$$\bar{p} = p_0 + A_0^+ E \bar{y}_0.$$

Notice that in each iteration, we have to compute a matrix-vector product of the form $(I + E^* \bar{R} A_0^+ E) \bar{y}_0$. Most of the work will be in computing the product $E^* \bar{R} A_0^+ E y$ for $y \in R^m$. Notice that here we do not have an explicit formula for $E^* \bar{R} A_0^+ E$ as in (2.3.17). If we use Alternative A of the last section, without taking advantage of the sparsity in $\bar{R}$, about $2n^3$ operations are required. In the single server case, this will be reduced to $2n^2 \log n$ operations. This is the same operation count as for solving the discrete Laplacian in the rectangular region by using the Fast Fourier Transformation. In Appendix A.2, we have

designed an algorithm that requires only $6n^2 + O(n)$ operations per iteration and $n^2 + O(n)$ memory. If one of the s_i equals one, we can reduce the counts to $5n^2 + O(n)$ operations and $O(n)$ memory spaces.

We remark that the matrix $\bar{R}$ is the discrete approximation of the operator which is zero in the rectangular region and has tangential derivatives along two of the sides. These sides correspond to states where one of the queue is full. Thus the matrix $\bar{A} = A_0 + \bar{R}$ resembles the finite difference approximation to the same continuous operator as discussed in previous sections, but with oblique derivatives on two of the sides and Neumann boundary conditions on the remaining ones. Numerical results show that the preconditioned system converges very fast, the number of iterations required to attain a given accuracy is roughly constant independent of n. This fast convergence can again be explained informally by the results in § 2.2.1.

In the next section, we will design other separable preconditioners for these networks by first perturbing the balance equations. Using the analogy, these preconditioners correspond to the finite difference approximation of the same operator but with a different type of boundary conditions.

§ 2.4 Other Separable Preconditioners

In this section, we will design other preconditioners for the models we discussed in § 2.3. We consider only separable preconditioners here because such systems can be solved economically. For simplicity, we confine ourselves to the model discussed in § 2.3.1. The idea can easily be extended to more general networks.

Let us assume that we are solving (2.3.2) - (2.3.4). Since A has a one dimensional null-space, we can fix one component of p, and solve the resulting non-singular system. More precisely, by lemma 2.1.1, p is positive, and is unique up to a multiple constant, thus we can always set $p_N = 1$ and partition the system (2.3.2) as

$$A\,p = \begin{bmatrix} B & d \\ c^* & \eta \end{bmatrix} \begin{bmatrix} \tilde{p} \\ p_N \end{bmatrix} = 0. \tag{2.4.1}$$

Using the facts that A is irreducible and has zero column sum, we see that B is irreducibly diagonally dominant and hence nonsingular. Thus we can proceed to solve the reduced system $B\,\tilde{p} = -\,d$ by direct or iterative methods, see Kaufman [25] and Funderlic and Mankin [17]. However, it is impossible to design a separable preconditioner for B because its dimension is $n_1 n_2 - 1$. To get around this, we can, instead of considering submatrices of A, consider a perturbed version of (2.3.2). More precisely, we fix p_N such that $\lambda_1 p_N = 1$, or equivalently, let $\lambda_1({}^1e_{n_1}\,{}^1e^*_{n_1} \otimes {}^2e_{n_2}\,{}^2e^*_{n_2})\,p = e_N$. We then obtain p from

$$\tilde{A}\,p = \{\,A + \lambda_1({}^1e_{n_1}\,{}^1e^*_{n_1} \otimes {}^2e_{n_2}\,{}^2e^*_{n_2})\,\}\,p = e_N. \tag{2.4.2}$$

Notice that $\tilde{A}$ is irreducibly diagonally dominant and therefore non-singular. Since $\tilde{A}$ is of order $n_1 n_2$, it is now possible to design separable preconditioners for $\tilde{A}$.

(I) A family of separable preconditioners for $\tilde{A}$

Let us partition $\tilde{A}$ as $\tilde{A} = \tilde{A}_1 + \tilde{R}_1$, where

$$\tilde{A}_1 = V_1 \otimes I_{n_2} + I_{n_1} \otimes G_2,$$
$$\tilde{R}_1 = \lambda_1 \cdot \{ {}^1e_{n_1} {}^1e_{n_1}^* \otimes \text{tridiag}\,(-1\,,\,0\,,\,0)\},$$
$$V_1 = G_1 + \lambda_1 \cdot {}^1e_{n_1} {}^1e_{n_1}^*. \tag{2.4.3}$$

Here the G_l are given by (2.1.6). Clearly $\tilde{A}_1$ is separable and V_1 is irreducible diagonally dominant. Hence $\tilde{A}_1$ is non-singular. We can write $p = \tilde{A}_1^{-1}\,\xi_1$ and solve the preconditioned system

$$\tilde{A}\,\tilde{A}_1^{-1}\,\xi_1 = (I + \tilde{R}_1\,\tilde{A}_1^{-1})\,\xi_1 = e_N.$$

Because of the sparsity of $\tilde{R}_1$, we can reduce this to an n_2 by n_2 system as in § 2.3.1. Unfortunately, numerical results show that the convergence rate for this preconditioned system is very slow. Notice that $\tilde{A}_1$ resembles the finite difference approximation of a second order elliptic operator on the square with a Dirichlet boundary condition on one of the side (see (2.4.3)) and Neumann boundary conditions on the remaining sides. In fact, if $\lambda_l = \mu_l = s_l = 1$ in (2.4.3), then $V_1 = \text{tridiag}\,(-1\,,\,2\,,\,-1) - e_1^*\,e_1$. This is exactly the finite difference approximation of a simple second order ordinary differential operator with a Neumann type data at one end and a Dirichlet type data at the other. Thus an intuitive explanation for the slow convergence is that $\tilde{A}_1$ is not a good approximation to $\tilde{A}$. It changes the oblique derivative in $\tilde{A}$ into a Dirichlet boundary condition.

Notice that $\tilde{A}_1$ is not the only non-singular separable preconditioner for $\tilde{A}$, in fact, there exists a family of non-singular separable preconditioners. Let us define, for any β,

$$V_\beta = G_1 + \beta\lambda_1 \cdot ({}^1e_{n_1} {}^1e_{n_1}^*), \tag{2.4.4}$$
$$\tilde{A}_\beta = V_\beta \otimes I_{n_2} + I_{n_1} \otimes G_2, \tag{2.4.5}$$
$$\tilde{R}_\beta = \lambda_1 \cdot \{ {}^1e_{n_1} {}^1e_{n_1}^* \otimes \text{tridiag}\,(-1\,,\,1-\beta\,,\,0)\}.$$

We note that $\tilde{A} = \tilde{A}_\beta + \tilde{R}_\beta$. Clearly $\tilde{A}_\beta$ is separable. When $\beta > 0$, $\tilde{V}_\beta$ is irreducibly diagonally dominant and hence $\tilde{A}_\beta$ is nonsingular. We can then define $p = \tilde{A}_\beta^{-1}\,\xi_\beta$, and solve for ξ_β. These preconditioners correspond to operators with a mixed type of boundary conditions on the side in question. Our numerical results show that the performance improves

when β gets closer to zero. This can again be explained informally by the analogy mentioned in § 2.2.1 and § 2.2.2.

Let us consider the case when $\beta = 0$. We obtain $V_0 = G_1$ and $\tilde{A}_0 = A_0$. This is the preconditioner considered previously. It is singular and corresponds to the operator with Neumann boundary conditions on every sides. Hence we cannot set $p = \tilde{A}_0^{-1} \xi_0$ and solve for ξ_0. However, we can still design a singular separable preconditioner for the non-singular matrix $\tilde{A}$, see (II) below. For $\beta < 0$, the preconditioners again correspond to operators with mixed type boundary conditions. Numerical results show that the convergence rate is slower when β becomes more negative. We remark that V_β in (2.4.4) can be symmetrized by a diagonal matrix, but it is no longer definite. More precisely, by the Cauchy interlace theorem (see Parlett [34]), and (2.4.4), V_β, and hence $\tilde{A}_\beta$, has one negative eigenvalue.

(II) Separable Preconditioner for $\tilde{A}$ when $\beta = 0$

By (2.1.25), there exists unique scalar α and $\xi_0 \in Im(A_0)$ such that $p = \alpha\, p_0 + A_0^+ \xi_0$. Since we have set $\lambda_1 p_N = 1$, α is no longer arbitrary. Thus besides solving for ξ_0, we need an extra equation for α. Since $\tilde{A} = A_0 + \tilde{R}_0$, we have

$$\tilde{A} (\alpha\, p_0 + A_0^+ \xi_0) = (I + \tilde{R}_0 A_0^+) \xi_0 + \alpha \tilde{R}_0 p_0 = e_N.$$

Since $\xi_0 \in Im(A_0)$, we have $1^* \xi_0 = 0$. This is the extra equation we need. We can now solve the following $(N+1)$ by $(N+1)$ system

$$F f = \begin{bmatrix} (I + \tilde{R}_0 A_0^+) & \tilde{R}_0 p_0 \\ 1^* & 0 \end{bmatrix} \begin{bmatrix} \xi_0 \\ \alpha \end{bmatrix} = \begin{bmatrix} e_N \\ 0 \end{bmatrix}. \tag{2.4.6}$$

We claim that F is non-singular. To prove this, suppose that $(\xi , \beta)^*$ is in the kernel of F. This implies that $\tilde{A} (\beta\, p_0 + A_0^+ \xi) = 0$ and $1^* \xi = 0$. Since $\tilde{A}$ is non-singular, the first condition implies that $\beta\, p_0 + A_0^+ \xi = 0$. The second condition implies that $\xi \in Im(A_0)$, which by the definition of A_0^+, implies that $A_0^+ \xi \in Im(A_0)$. Thus by (2.1.20), we have $\beta = 0$ and

$A_0^+ \xi = 0$. By the invertability of A_0^+ on $Im(A_0)$, we have $\xi = 0$. Hence F is non-singular and it is legitimate to solve for f in (2.4.6). By the sparsity of $\tilde{R}_0$, we can reduce (2.4.6) to an (n_2+1) by (n_2+1) system.

Notice that by (2.4.2), $\tilde{A}$ differs from A by a rank one matrix. Thus $\tilde{R}_0$ differs from the R_0 in (2.3.5) by a rank one matrix. Hence the N by N leading sub-matrix of $F\,F^*$ differs from the preconditioned matrix $(I+R_0\,A_0^+)\,(I+R_0\,A_0^+)^*$ considered in § 2.3.1 by at most a rank three matrix. Using the Cauchy interlace theorem, the singular values of $(I+R_0\,A_0^+)$ will interlace the singular values of F, except possibly a few outlying ones. In particular, if the singular values of $(\,I + R_0\,A_0^+\,)$ are clustered, so will be the singular values of F. The numerical results in § 4 show that the convergence rate for these two systems are very much the same.

§ 2.5 General Overflow Queueing Models

The idea of preconditioning the singular generating matrix A by another singular matrix A_0 can easily be extended to more general overflow queueing networks where A and A_0 have the same order. Let us first illustrate how to apply our method to the 3-queue model discussed in Kaufman [25]. We then generalize our method to a family of overflow queueing models with $q > 3$.

§ 2.5.1 A Three-queue Overflow Model

Consider the following 3-queue network. Customers entering or being overflowed into the i-th queue, $i = 1, 2$, will be overflowed and served at the $(i+1)$-th queue if all the spaces in the i-th queue are occupied. Customers entering the third queue will be lost if the third queue is filled. Moreover, once a customer entered a queue, he will stay in that queue until he is served. In particular, customers cannot jump between the waiting lines. The Kolmogorov equations of this network are given by

$$\begin{aligned}
&\Big\{\lambda_1(1-\delta_{i,n_1-1}\delta_{j,n_2-1}\delta_{k,n_3-1}) + \lambda_2(1-\delta_{j,n_2-1}\delta_{k,n_3-1}) + \lambda_3(1-\delta_{k,n_3-1}) \\
&\quad + \mu_1 \min(i,s_1) + \mu_2 \min(j,s_2) + \mu_3 \min(k,s_3) \Big\} p_{ijk} \\
&= \lambda_1(1-\delta_{i,0})\, p_{i-1jk} + (1-\delta_{j,0})(\lambda_1\delta_{i,n_1-1}+\lambda_2)\, p_{ij-1k} \qquad (2.5.1) \\
&\quad + (1-\delta_{k,0})(\lambda_1\delta_{i,n_1-1}\delta_{j,n_2-1}+\lambda_2\delta_{j,n_2-1}+\lambda_3)\, p_{ijk-1} + \mu_1 \min(i+1,s_1)(1-\delta_{i,n_1-1})\, p_{i+1jk} \\
&\quad + \mu_2 \min(j+1,s_2)(1-\delta_{j,n_2-1})\, p_{ij+1k} + \mu_3 \min(k+1,s_3)(1-\delta_{k,n_3-1})\, p_{ijk+1},
\end{aligned}$$

where $0 \le i < n_1$, $0 \le j < n_2$ and $0 \le k < n_3$. Thus there are $N = n_1\, n_2\, n_3$ states in the system. The corresponding matrix equation is given by

$$A\, p = (A_0 + R_0)\, p = 0. \qquad (2.5.2)$$

Here A_0 is given by (2.1.17) and

$$R_0 = {}^1e_{n_1}\,{}^1e_{n_1}^* \otimes {}^2R_1 \otimes I + I \otimes {}^2e_{n_2}\,{}^2e_{n_2}^* \otimes {}^3R_2 + {}^1e_{n_1}\,{}^1e_{n_1}^* \otimes {}^2e_{n_2}\,{}^2e_{n_2}^* \otimes {}^3R_1, \quad (2.5.3)$$

where ${}^jR_l$ are given by (2.3.6). Since p is a probability distribution, we further require that

$$1^* p = 1, \quad (2.5.4)$$

$$p_{ijk} \geq 0. \quad (2.5.5)$$

The terms in (2.5.3) indicate the direction of the overflow. The first term corresponds to the overflow of customers from queue 1 to queue 2, the second term corresponds to the overflow from queue 2 to queue 3 and the third term corresponds to the case where queue 1 and queue 2 are filled and customers from queue 1 are overflowed to queue 3. We note that R_0 is sparse. It has only $m = (n_1 + n_2 - 1)\, n_3$ rows with non-zero entries, and that every such row has at most three non-zero entries.

As in § 2.3, we can transform (2.5.2) into an inhomogeneous system

$$(I + R_0 A_0^+)\, \xi_0 = -R_0 p_0, \quad (2.5.6)$$

where $\xi_0 \in Im(A_0)$ and

$$p = p_0 + A_0^+ \xi_0. \quad (2.5.7)$$

By using (2.5.3), it is easily checked that $R_0 p_0 \neq 0$. By the form of ${}^jR_l$ in (2.3.6), it is also clear that $A = A_0 + R_0$ satisfies the assumptions of lemma 2.1.1, hence the solution to (2.5.2), (2.5.4) and (2.5.5) exists and is unique. Thus by lemma 2.3.1, we see that the matrix in (2.5.6) is non-singular. Since R_0 is sparse and has m non-zero rows, we can reduce this system to an m by m system. Notice that we need $O(m^2) = O(n_l^4)$ storage spaces to hold the entries of this matrix. For a network with $n_l = 20$, the storage requirement is about 640,000.

Using iterative methods discussed in § 2.2 will require much less storage. The cost per iteration depends mostly on the cost of the matrix-vector multiplication. If we use (2.1.22) to compute $A_0^+ \xi$ for any vector ξ, it is clear that this computation require $2\, n_1 n_2 n_3 (n_1 + n_2 + n_3) + O(n_l^3)$ operations. By using the sparsity of R_0, $R_0 \xi$ can be computed in $3\, m = O(n_l^2)$ operations. Thus assuming that $n_1 = n_2 = n_3 = n$, the cost per

iteration is $6n^4 + O(n^3)$ operations. The storage requirement is $n^3 + O(n^2)$. In the single-server case, we can reduce the operation counts of our method to $O(n^3 \log n)$. Let us point out that by using the sparsity of R_0, we can actually reduce the counts to $O(n^2)$ storage spaces and $O(n^3)$ operations per iteration. This algorithm is given in Appendix A.3.

We remark that by using (2.5.3) and the form of ${}^jR_l$ in (2.3.6), the operator R_0 resembles an operator which is zero in a cube and has tangential derivatives on two of the faces. These particular faces correspond to those states where either queue 1 or queue 2 is full. Recall that A_0 resembles the standard seven point operator that obtained when discretizing an elliptic operator in a cube with Neumann type boundary conditions everywhere. Thus the matrix $A = A_0 + R_0$ resembles A_0 but with oblique derivatives on the two particular faces. Hence under this analogy, A_0 and A has the similar boundary conditions. From our previous experience, we expect that A_0 to be a good preconditioner in this case and this fact is confirmed by the numerical results in § 4.

Next we extend our method to a more general type of overflow models which include this model and the models we discussed in § 2.3 as particular cases.

§ 2.5.2 A q-Queue Overflow Model

Let us consider the following q-queue networks. Customers entering into a full queue will be overflowed to another queue or be blocked and lost. To avoid ambiguity, we assume that for any given queue there is at most one direction of overflow of customers. However, we permit successive overflowing, i.e., customers being overflowed from the i-th queue to the j-th queue say, can be overflowed to another queue if the j-queue is also filled. To prevent customers from wandering within the network, we assume that any given path of successive overflowing does not form a loop. We assume moreover that, once a customer has entered a queue, he will wait in that queue until he is served. In particular, customers cannot jump between the waiting lines and the overflow of customers from any queue can occur only

when the queue is full. These conditions will ensure that all the N states in the network are admissible. Hence the generating matrix A will be of the same order as the preconditioner A_0. As in § 2.5.1, we have

$$A = A_0 + R_0, \tag{2.5.8}$$

where R_0 consists of terms like those in (2.5.3). To be more specific, let us use the notations $E_l = {}^l e_{n_l} {}^l e_{n_l}^*$ and $E_l^0 = {}^l R_j^0 = I$ where the matrices ${}^l R_j$ are given by (2.3.6). If customers entering the i-queue will be overflowed and served at the j-queue when the i-queue is full then R_0 contains the following term

$$R_{ij} = \bigotimes_{k=1}^{q} E_l^{\delta_{ik}} {}^l R_l^{\delta_{jk}}. \tag{2.5.9}$$

This is because overflow from the i-th queue to the j-th queue corresponds to adding

$$\delta_{k_i n_i - 1} (1 - \delta_{k_j n_j - 1}) \lambda_l p_{k_1, \ldots, k_q} \tag{2.5.10}$$

to one side of the Kolmogorov balance equations and

$$(1 - \delta_{0k_j}) \delta_{k_i n_i - 1} \lambda_l p_{k_1, \ldots, k_j - 1, \ldots, k_q} \tag{2.5.11}$$

to the other side, for $1 \le k_j < n_j$, $1 \le j \le q$. The term in (2.5.10) indicates that we are leaving the state $(k_1, \ldots, k_q)$ at an additional rate λ_l when the i-th queue is full and the j-th queue is not yet full. The term in (2.5.11) indicates that we are entering the state $(k_1, \ldots, k_q)$ at an additional rate λ_l from the state $(k_1, \ldots, k_{j-1}, \ldots, k_q)$ when the i-th queue is full. Successive overflow from one queue to another means that the term added to R_0 has more E_l factors in it. For example, if overflow from the i-th queue to the j-th queue can be overflowed to the k-th queue when the j-th queue is filled, then we add to R_0 the term

$$R_{ijk} = \bigotimes_{l=1}^{q} E_l^{\delta_{il}} E_j^{\delta_{jl}} {}^k R_l^{\delta_{kl}}. \tag{2.5.12}$$

This follows from the fact that we have to add

$$\delta_{l_i n_i - 1} \delta_{l_j n_j - 1} (1 - \delta_{l_k n_k - 1}) \lambda_l p_{l_1, \ldots, l_q}$$

and

$$(1-\delta_{0l_i})\,\delta_{l_i n_i-1}\,\delta_{l_j n_j-1}\,\lambda_l\,p_{l_1,\ldots,l_i-1,\ldots,l_q}$$

for $1 \le l_j < n_j$, $1 \le j \le q$, to the two sides of the Kolmogorov balance equations. Notice that every terms added to R_0 has one and only one ${}^jR_l$ factor in it. Here the i and j indicate the origin and the final destination of the overflowed customers respectively.

Notice that by (2.3.6) and the definition of E_l, all the terms added to R_0 have zero column sums, nonnegative diagonal and nonpositive off-diagonal entries. Thus $A = A_0 + R_0$ still satisfies the assumptions of lemma 2.1.1. Hence the steady-state probability distribution exists and is unique. Thus by lemma 2.3.1, the system $(I + R_0 A_0^+)$ is non-singular. Next we notice that those terms added to R_0 are sparse. In fact, since overflow occurs only when at least one of the queues, say the i-th queue, is full, the corresponding term has at most N/n_l non-zero rows. Moreover, since every such term represents the overflow from one queue to another, every non-zero rows has at most two non-zero entries. If we permit all possible directions of overflow within the network, the total number of non-zero rows in R_0 will be bounded above by

$$m_q = N \cdot \sum_{l=1}^{q} \frac{1}{n_l}. \tag{2.5.13}$$

Moreover, the total number of non-zero entries in every such rows will not exceed $q + 1$, since there are only q queues in the network. Thus R_0 is also sparse.

Using the sparsity of R_0, we can reduce the system $(I + R_0 A_0^+)$ to a system which has order at most m_q. However, for general network, this will often be too large to handle by direct methods. For these methods would require $O(m_q^2) = O(n^{2q-2})$ storage spaces. Notice that for any given vector ξ, the vector-matrix multiplication $R_0\,\xi$ requires at most $(q+1)\,m_q$ operations. This work is negligible when compared to the work of computing $A_0^+\,\xi$ by (2.1.22), which is of $O(n^{q+1})$. Thus using iterative methods mentioned in § 2.2, the work and storage required for each iteration are roughly $O(n^{q+1})$ and $O(n^q)$ respectively.

Notice that, using the continuous analogy, each term added to R_0 corresponds to a forward difference operator on a particular face in the N-cube. These faces correspond to

states where overflow occurs. Thus R_0 corresponds to an operator which is zero in the N-cube, but with tangential derivatives on some of the faces. Recall that A_0 corresponds to an elliptic operator acting on this N-cube with Neumann boundary conditions everywhere. Thus the matrix $A = A_0 + R_0$ resembles the same operator but with oblique derivatives on those particular faces. Hence the boundary conditions of A and A_0 are of the same type. From the experience in previous sections, we expect our method to have fast convergence for these kind of models.

In the next section, we consider models in which the dimension of the state space, i.e., the dimension of the generating matrix A, is less than the dimension of A_0.

§ 2.6 An Overflow Model with Restricted State-space

In this section, we consider a model in which overflow occurs even before a queue is full. The resulting problem is still a homogeneous system of the form $A\,p = 0$, but with some of the entries of p set to zero. Thus the dimension N_A of A is less than the dimension N of the preconditioner A_0 discussed in § 2.1. We introduce two methods here to solve this queueing problem. In the first, we partition the state space into subspaces in which we can find separable preconditioners. In the second, we embed the state space into one where we can use A_0 as a preconditioner.

Let us consider the following 2-queue network. Customers entering the first queue will wait and be served at the second queue if all the spaces in the first queue is filled. Moreover, we assume that a customer waiting for service in the first queue is moved to and served at the second queue if a server in the second queue becomes available. Thus, some of the states are not permissible here. More precisely, we have

$$p_{I,J} = 0 \qquad s_1 < i < n_1,\ 0 \le j < s_2. \tag{2.6.1}$$

We may associate the values of $p_{I,J}$ with the following L-shaped region:

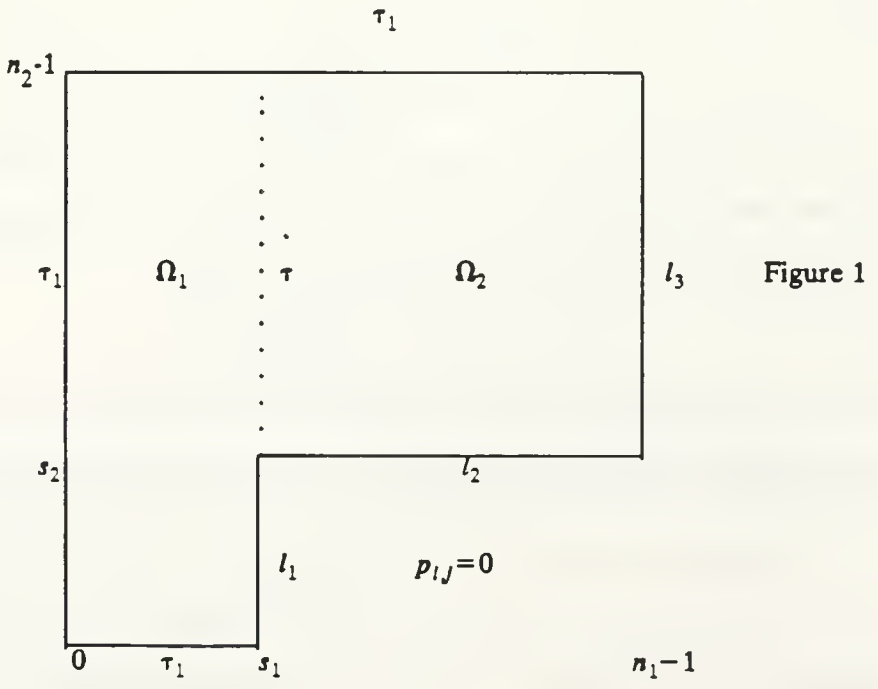

Figure 1

In the figure, l_1, l_2, l_3 are line segments defined by

$$l_1 = \{ s_1 \} \times [\, 0 \,,\, s_2 - 1 \,], \tag{2.6.2}$$

$$l_2 = [\, s_1 + 1 \,,\, n_1 - 1 \,] \times \{ s_2 \}, \tag{2.6.3}$$

$$l_3 = \{ n_1 - 1 \} \times [\, s_2 \,,\, n_2 - 2 \,]. \tag{2.6.4}$$

For simplicity, we let

$$\tau_2 = l_1 \cup l_2 \cup l_3 \cup \{ (s_1,s_2) \,,\, (n_1-1,n_2-1) \}. \tag{2.6.5}$$

We remark that this model is similar to the one discussed in Kaufman, Serry and Morrison [24], except that we have added a feature. Namely, we permit overflow from the first queue to the second queue when the first queue is full. The Kolmogorov balance equations of our model are given by

$$\begin{aligned}
&[\lambda_1 (1-\delta_{i n_1-1}\delta_{j n_2-1}\chi_{j-s_2}) + \lambda_2(1-\delta_{j n_2-1}) + \min(i,s_1)\mu_1 + \min(j,s_2)\mu_2 \,]\, p_{ij} \\
&\quad = (1-\chi_{i-1-s_1}\chi_{s_2-1-j})\, [\lambda_1(1-\delta_{i0})\, p_{i-1j} + (1-\delta_{j n_2-1})\, \min(j+1,s_2)\, \mu_2\, p_{ij+1}] \\
&\quad + (1-\delta_{j0})\, [\lambda_1(\delta_{i s_1}\chi_{s_2-j} + \delta_{i n_1-1}\chi_{j-s_2}) + \lambda_2(1-\chi_{i-1-s_1}\chi_{s_2-j})]\, p_{ij-1} \\
&\quad + (1-\delta_{i n_1-1})\, [(1-\chi_{i-s_1}\chi_{s_2-1-j})\, \min(i+1,s_1)\mu_1 + s_2\mu_2\chi_{i-s_2}\delta_{j s_2}]\, p_{i+1j},
\end{aligned} \tag{2.6.6}$$

for $0 \le i < n_1$ and $0 \le j < n_2$. Here

$$\chi_l = \begin{cases} 1, & l \ge 0, \\ 0, & l < 0. \end{cases} \tag{2.6.7}$$

We note that these equations imply (2.6.1). From (2.6.6), we see that the steady-state probability distribution p satisfies the homogeneous equation

$$A\, p = 0. \tag{2.6.8}$$

Here A is the generating matrix of dimension N_A and satisfies the assumptions of lemma 2.1.1. Since p is a probability distribution, we supplement (2.6.8) with

$$1^* p = 1, \tag{2.6.9}$$

$$p_{ij} \ge 0. \tag{2.6.10}$$

By lemma 2.1.1, the solution p to (2.6.8) - (2.6.10) exists and is unique. Moreover,

$$p_{ij} > 0. \tag{2.6.11}$$

Considering the continuous analogy, we find that the matrix A resembles an elliptic operator acting on the L-shaped region with Neumann boundary conditions on τ_1 and oblique boundary conditions on τ_2. The idea of the previous sections would suggest the partition

$$A = \bar{A} + \bar{R}, \tag{2.6.12}$$

with $\bar{A}$ resembling the same operator but with Neumann boundary conditions everywhere. $\bar{R}$ will then be an operator that is zero in the L-shape region, but has tangential derivatives along τ_2. We note that $\bar{A}$ has the form

$$\bar{A} = \begin{bmatrix} T_1 & D_1 & \mathbf{0} \\ E_1 & C_1 & D_2 \\ \mathbf{0} & E_2 & T_2 \end{bmatrix}. \tag{2.6.13}$$

Here T_l represents couplings between the pairs of states in Ω_l, C_1 couplings between the pairs of states on the interface τ, and D_l and E_l couplings between the pairs belonging to Ω_l and τ. The dimension of C_1 is equal to the number of states on τ, which is equal to $n_2 - s_2$. This is small when compared to the dimension of T_l, which is equal to the total number of states in Ω_l. The dimension of T_1 is $s_1 n_2$ and that of T_2 is $(n_2 - s_2)(n_1 - s_1 - 1)$. We note that in (2.6.13)

$$E_2 = [\, -\lambda_1 \cdot I_{n_2 - s_2}, \mathbf{0} \,]^*, \tag{2.6.14}$$

$$D_2 = [\, -s_1 \mu_1 \cdot I_{n_2 - s_2}, \mathbf{0} \,], \tag{2.6.15}$$

where $\mathbf{0}$ is the zero matrix of order $(n_2 - s_2)$ by $(n_1 - s_1 - 2)$. Thus they are sparse.

The exact form of $\bar{A}$ can be obtained by using the equations given in § 2.6.1. More precisely, by (2.6.13)

$$\bar{A} = \begin{bmatrix} T_1 & D_1 & \mathbf{0} \\ E_1 & C_2 & D_2 \\ \mathbf{0} & \mathbf{0} & T_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C_1 - C_2 & \mathbf{0} \\ \mathbf{0} & E_2 & \mathbf{0} \end{bmatrix}, \tag{2.6.16}$$

where the submatrices E_2, D_2, $C_1 - C_2$, $\tilde{A}_0 = \begin{bmatrix} T_1 & D_1 \\ E_1 & C_2 \end{bmatrix}$ and T_2 are given by (2.6.14), (2.6.15), (2.6.26), (2.6.29) and (2.6.30) respectively. It is then easy to check that $\mathbf{1}^* \bar{A} = \mathbf{0}$. Since $\bar{R} = A - \bar{A}$ and $\mathbf{1}^* A = \mathbf{0}$, we have

$$1^* \bar{R} = 1^* \bar{A} = 0. \tag{2.6.17}$$

We claim that the matrix $\bar{R}$ is sparse. We first note that, depending on the ordering of the states, any index j, $1 \le j \le N_A$, corresponds to a unique state (j_1, j_2) in the L-shaped domain. Using this notation, it is straightforward to check that

$$\bar{R} = \bar{R}_1 + \bar{R}_2. \tag{2.6.18}$$

Here $\bar{R}_1$ is a diagonal matrix given by

$$(\bar{R}_1)_{jj} = \begin{cases} \lambda_1 + s_2 \mu_2 & (j_1, j_2) = (n_1 - 1, s_2), \\ \lambda_1 & (j_1, j_2) \in l_1 \cup l_3 \setminus \{(n_1 - 1, s_2)\}, \\ s_2 \mu_2 & (j_1, j_2) \in l_2 \setminus \{(n_1 - 1, s_2)\}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6.19}$$

and $\bar{R}_2$ is given by

$$(\bar{R}_2)_{kj} = \begin{cases} -\lambda_1 & (k_1, k_2 - 1) \text{ and } (j_1, j_2) \in l_1 \cup l_3, \\ -s_2 \mu_2 & (k_1 + 1, k_2) \text{ and } (j_1, j_2) \in l_2, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6.20}$$

Thus the number of non-zero rows is equal to the number of states on τ_2, which is equal to

$$N_{\bar{R}} = n_1 + n_2 - s_1 - 1. \tag{2.6.21}$$

Moreover, every such rows has at most two non-zero entries. It can be shown that $\bar{A}$ still satisfies the assumptions of lemma 2.1.1. Hence it is singular with a one dimensional null-space. However, $\bar{A}$ is not separable. Hence $\bar{A}^+ x$ cannot be computed economically. In the following, we will design singular, separable preconditioners that are close to A in the sense that they represent the same operator in the L-shaped region and have the same type of boundary conditions. The first idea comes from the theory of substructuring.

§ 2.6.1 Partitioning of the State-space

The method of substructuring have been used for solving elliptic problems defined in irregular regions, see Bjørstad and Widlund [7], Dryja [14], Bramble et al. [8], Buzbee et al. [10] and Concus et al. [11]. The idea is to partition the problem into subproblems which

correspond to subregions into which the original region has been partitioned. We can then solve each of these subproblems separately by direct methods while the interactions between the subregions are solved by a direct or iterative method. Since the number of nodes on the interface usually is small compared to the number of nodes in each subregions, the size of this interface problem is usually small compared to the original problem. If we are using iterative methods to solve the interface problem, then in each iteration, we will have to solve the subproblems once in each subregion. However, if the boundary conditions for the original region are such that separation of variables is possible, then solving the subproblems by direct methods will require very little amount of work.

To be more specific, let us consider the problem of solving Laplace's equation in the L-shaped region depicted in Figure 1, with Neumann boundary conditions everywhere. The theory of substructuring suggests the following preconditioner. We first solve the problem defined on Ω_1 with Neumann boundary conditions on the boundary of Ω_1 including τ. This is a separable problem. Having solved this problem, we use the value of the solution on the interface τ as Dirichlet data and solve a Dirichlet-Neumann problem on Ω_2 with Dirichlet boundary condition on the interface τ and Neumann boundary conditions on the remaining three sides. This problem is also separable.

Using the analogy between the queueing model and this continuous problem, we construct our preconditioner accordingly. The numerical results in § 4 show that this preconditioner is very good. In matrix terms, we partition our matrix A as

$$A = \tilde{A} + \tilde{R}, \tag{2.6.22}$$

where

$$\tilde{A} = \begin{bmatrix} \tilde{A}_0 & \mathbf{0} \\ \tilde{E}_2 & T_2 \end{bmatrix} = \begin{bmatrix} T_1 & D_1 & \mathbf{0} \\ E_1 & C_2 & \mathbf{0} \\ \mathbf{0} & E_2 & T_2 \end{bmatrix}. \tag{2.6.23}$$

Here

$$\tilde{E}_2 = [\, \mathbf{0} \,, E_2 \,], \tag{2.6.24}$$

and

$$\tilde{A}_0 = \begin{bmatrix} T_1 & D_1 \\ E_1 & C_2 \end{bmatrix} \tag{2.6.25}$$

with

$$C_2 = C_1 - \lambda_1 \cdot I_{n_2 - s_2}. \tag{2.6.26}$$

By (2.6.12) and (2.6.16),

$$\tilde{R} = \bar{R} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_1 - C_2 & D_2 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.6.27}$$

Since $\bar{R}$, D_2 and $C_1 - C_2$ are sparse, $\tilde{R}$ is also sparse. In fact it has at most

$$m = n_1 + 2\,n_2 - s_1 - s_2 - 2 \tag{2.6.28}$$

non-zero rows, and every such rows has at most two non-zero entries. Notice that the sub-matrix $\tilde{A}_0$ corresponds to a Neumann problem on the subregion Ω_1. In fact,

$$\tilde{A}_0 = \tilde{G}_1 \otimes I_{n_2} + I_{s_1+1} \otimes G_2, \tag{2.6.29}$$

where $\tilde{G}_1$, of dimension s_1+1, is the same as G_l in (2.1.6) but with s_1 replacing $n_l - 1$ there. Thus $\tilde{A}_0$ is the generating matrix of the free model discussed in § 2.1 with s_1 spaces in the first queue and $n_2 - 1$ spaces in the second queue. Hence by lemma 2.1.2, $\tilde{A}_0$ is separable, has a one dimensional null-space and a positive null-vector. Let us denote its null-vector by $\tilde{p}_0$.

On the other hand, T_2 corresponds to a mixed type problem defined on Ω_2. In fact,

$$T_2 = V_1 \otimes I_{n_2 - s_2} + I_{n_1 - s_1 - 1} \otimes V_2. \tag{2.6.30}$$

Here

$$V_1 = \text{tridiag}\,(\,\lambda_1\,,\,\lambda_1 + s_1\,\mu_1\,,\,s_1\,\mu_1\,) - \lambda_1 \cdot e_{n_1 - s_1 - 1}\,e^*_{n_1 - s_1 - 1} \tag{2.6.31}$$

is a matrix of order $n_1 - s_1 - 1$ and

$$V_2 = \text{tridiag}\,(\,\lambda_2\,,\,\lambda_2 + s_2\,\mu_2\,,\,s_2\,\mu_2\,) - s_2\,\mu_2 \cdot e_1\,e_1^* - \lambda_2 \cdot e_{n_2 - s_2}\,e^*_{n_2 - s_2} \tag{2.6.32}$$

is a matrix of order $n_2 - s_2$. It is clear that T_2 is separable and since V_1 is irreducibly diagonally dominant, T_2 is non-singular. Thus by (2.6.23), $\tilde{A}$ is singular, and has a one dimensional null-space. The null-vector of $\tilde{A}$ is given by

$$\tilde{p} = \begin{bmatrix} \tilde{p}_0 \\ -T_2^{-1}\,\tilde{E}_2\,\tilde{p}_0 \end{bmatrix}. \tag{2.6.33}$$

We note that the $\tilde{p}_{ij}$ are not necessarily positive. Let us define the generalized inverse $\tilde{A}^+$ of $\tilde{A}$ as

$$\tilde{A}^+ = \begin{bmatrix} \tilde{A}_0^+ & 0 \\ -T_2^{-1}\tilde{E}_2\tilde{A}_0^+ & T_2^{-1} \end{bmatrix}. \tag{2.6.34}$$

Similarly to lemma 2.1.2, we have

LEMMA 2.6.1

(i) $Im(\tilde{A}) = \{x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in R^{N_A} \mid x_1 \in Im(\tilde{A}_0)\}$.

(ii) $\tilde{A}^+$ is invertible on $Im(\tilde{A})$. More precisely, for all $y \in Im(\tilde{A})$, there exists a unique $x \in Im(\tilde{A})$, such that

$$\tilde{A}^+ x = y, \tag{2.6.35}$$

where $x = \tilde{A}\, y$. Thus for all $y \in Im(\tilde{A})$,

$$\tilde{A}\,\tilde{A}^+ y = \tilde{A}^+\tilde{A}\, y = y. \tag{2.6.36}$$

(iii) For all $y \in R^{N_A}$, there exists a unique α and $\xi \in Im(\tilde{A})$ such that,

$$y = \alpha\,\tilde{p} + \tilde{A}^+\xi. \tag{2.6.37}$$

(iv) Let p be the solution to (2.6.8) - (2.6.10), then there exists a unique $\alpha \neq 0$ and $\xi \in Im(\tilde{A})$ such that,

$$p = \alpha\,\tilde{p} + \tilde{A}^+\xi. \tag{2.6.38}$$

Proof: The proof of (i) - (iii) follows easily from lemma 2.1.2 and the fact that T_2 is non-singular. Thus let us prove (iv). By (iii), it suffices to show that $\alpha \neq 0$. Suppose $\alpha = 0$, then $p = \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} = \tilde{A}^+\xi \in Im(\tilde{A})$. By (i), $p_0 \in Im(\tilde{A}_0)$, hence $1^* p_0 = 0$, contradicting (2.6.11). □

We remark that even though $\tilde{A}$ cannot be symmetrized, we still have a decomposition of the

state space as in (2.6.37). Notice that in (2.6.23), $\tilde{A}$ is in block lower triangular form. Since $\tilde{A}_0$ and T_2 are separable, and by (2.6.14) and (2.6.24), $\tilde{E}_2$ is sparse, thus $\tilde{A}^+ x$ can be computed easily for any $x \in Im(\tilde{A})$. We remark that $\tilde{E}_2$ picks up the Dirichlet data on τ.

Using (iv) in lemma 2.6.1 and since p is unique up to a multiple constant, we may let

$$p = \tilde{p} + \tilde{A}^+ \xi_0, \tag{2.6.39}$$

and normalize it by (2.6.9) after we find it. Putting this expression into (2.6.8), we get

$$(I + \tilde{R}\,\tilde{A}^+)\xi_0 = -\tilde{R}\,\tilde{p}. \tag{2.6.40}$$

It can easily be checked that $\tilde{R}\,\tilde{p} \neq 0$. Notice that in the splitting (2.6.22), we do not have $1^* \tilde{A} = 0$. Thus $Im(A)$, and hence $Im(\tilde{R})$, are not necessarily in $Im(\tilde{A})$. Hence we cannot draw the same conclusion as in lemma 2.3.1 that the matrix $(I + \tilde{R}\,\tilde{A}^+)$ is nonsingular. However, it is clear that this matrix maps $Im(\tilde{R})$ into itself. By (2.6.40), it is also clear that $\xi_0 \in Im(\tilde{R})$. By the existence and uniqueness of ξ_0, the matrix is invertible in $Im(\tilde{R})$. Thus we may proceed to solve for ξ_0 by considering $(I + \tilde{R}\,\tilde{A}^+)|_{Im(\tilde{R})}$. As mentioned in § 2.2, if we use the Orthodir method to solve this kind of equations, then there is no need to do the restriction explicitly. We remark that since $\tilde{R}$ has m non-zero rows, where m is given by (2.6.28), (2.6.40) is practically an m by m system.

Let us calculate the cost of computing $\tilde{R}\,\tilde{A}^+ x$ for any x. Since $\tilde{R}$ has at most $2m$ non-zero entries, $\tilde{R}\,x$ can be computed in $2m$ operations. Next let us consider solving

$$\tilde{A} \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix},$$

where $x_0 \in Im(\tilde{A}_0)$. Notice that by (2.6.23), $\tilde{A}_0\, y_0 = x_0$. In view of (2.6.29), this system can be solved by the Alternative A of § 2.3.1 in $2n_2^2$ operations. Let us remark that when $s_1 << n_2$, Alternative B in Appendix A.1 requires only $4n_2 s_1$ operations. Having found y_0, we solve $T_2\, y_1 = x_1 - \tilde{E}_2\, y_0$. Since $\tilde{E}_2$ is sparse, the right hand side of this equation can be computed in $n_2 - s_2$ operations. Notice that in (2.6.30), T_2 is separable. Thus y_1 can be solved by first diagonalizing V_2 in (2.6.32) and then solving the resulting tridiagonal systems; see Alternative B in Appendix A.1 for more detail. We note that we can use the Fast Fourier

transform to perform the diagonalization. This follows from the fact that V_2 is the same as the G_l in (2.3.20), but with the service rate $s_2\ \mu_2$ rather than μ_l. Thus the work is roughly

$$5\ (n_2-s_2)(n_1-s_1) + 2\ (n_2-s_2)\log\ (n_2-s_2).$$

The first term here is the work required to solve the resulting tridiagonal systems. Combining these results, we see that the work required in the j-th iteration is

$$4\ n_2 s_1 + 5\ (n_2-s_2)(n_1-s_1) + 2\ (n_2-s_2)\log\ (n_2-s_2) + 3\ jm, \tag{2.6.41}$$

and the memory requirement is

$$2jm\ \div\ s_1^2 + O(n_l-s_l). \tag{2.6.42}$$

We note that there are many other viable separable preconditioners. For example, instead of solving the Dirichlet-Neumann problem corresponding to T_2, we may solve a Dirichlet problem on Ω_2. This is also a separable problem. However, from the discussion in § 2.2.1 and in Bjørstad and Widlund [7], we expect that this preconditioner will not lead to an optimal method.

§ 2.6.2 Embedding of the State-space

Capacitance matrix methods have been developed for solving elliptic problems in irregular regions such as the L-shaped region in Figure 1, see O'Leary and Widlund [32], Proskurowski and Widlund [35] and Astrakhantsev [1]. The idea is to embed the state-space into a larger space where there is a separable preconditioner. Here we design an algorithm that adopt this approach.

Recall that in § 2.1, A_0 resembles a Neumann problem on the whole rectangular region $[0,\ n_1-1] \times [0,\ n_2-1]$. If we order the states in the L-shaped region first, then we can write

$$A_0 = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{2.6.43}$$

where A_{11} and A_{22} are square matrices of dimension N_A and $N - N_A$ respectively. We claim that A_{22} is nonsingular. In fact, by the definition of A_0,

$$A_{22} = V_1 \otimes I_{s_2} + I_{n_1 - s_1 - 1} \otimes \tilde{G}_2, \tag{2.6.44}$$

where V_1 is given by (2.6.31) and $\tilde{G}_2$, of dimension s_2, is the principal submatrix of G_2 obtained by deleting the last $n_2 - s_2$ columns and rows of G_2. Since V_1 and $\tilde{G}_2$ are irreducibly diagonally dominant, A_{22} is nonsingular.

Consider the N by N matrix

$$A_N = \begin{bmatrix} A & A_{12} \\ 0 & A_{22} \end{bmatrix}, \tag{2.6.45}$$

where A is the generating matrix in (2.6.8). Clearly if p is the solution to (2.6.8) - (2.6.10), then $p_N = \begin{bmatrix} p \\ 0 \end{bmatrix}$ is the unique solution to

$$A_N p_N = 0, \tag{2.6.46}$$

$$1_N^* p_N = 1, \tag{2.6.47}$$

and

$$(p_N)_{kj} \geq 0, \quad 1 \leq k, j \leq N. \tag{2.6.48}$$

Here $1_N = (1, 1, \ldots, 1) \in R^N$. Since $1_N^* A_0 = 0$ and $1^* A = 0$, it follows that

$$1_N^* A_N = 0. \tag{2.6.49}$$

Moreover, by (2.6.47) and (2.1.25), there exists a unique $\xi_0 \in Im(A_0)$, such that

$$p_N = p_0 + A_0^+ \xi_0, \tag{2.6.50}$$

where p_0 is given by (2.1.11). Define

$$R_N = A_N - A_0 = \begin{bmatrix} A - A_{11} & 0 \\ -A_{21} & 0 \end{bmatrix}. \tag{2.6.51}$$

Then (2.6.46) and (2.6.50) imply that

$$(I + R_N A_0^+) \xi_0 = -R_N p_0. \tag{2.6.52}$$

By (2.6.49) and lemma 2.3.1, the matrix $(I + R_N A_0^+)$ is nonsingular. Thus we can solve (2.6.52) either by direct or iterative methods.

We claim that the matrix R_N is sparse. In fact, by the definition of A_0

$$(A_{21})_{kj} = \begin{cases} -\lambda_1 & (k_1 - 1, k_2) \text{ and } (j_1, j_2) \in l_1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6.53}$$

On the other hand, it is straightforward to check that

$$A_{11} = \bar{A} + D, \tag{2.6.54}$$

where $\bar{A}$ is given by (2.6.16) and D is a diagonal matrix such that

$$D_{jj} = \begin{cases} \lambda_1 & (j_1, j_2) \in l_1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6.55}$$

By (2.6.12) and (2.6.54), $A - A_{11} = \bar{R} - D$. From (2.6.18) - (2.6.20), we see that $\bar{R} - D$ still has N_R nonzero rows and every such rows has at most two nonzero entries. Here N_R is given by (2.6.21). Moreover, by (2.6.51) and (2.6.53), the number of nonzero rows in R_N is equal to

$$N_R = N_{\bar{R}} + s_2 = n_1 + n_2 + s_2 - s_1 - 1, \tag{2.6.56}$$

and every such rows has at most two nonzero elements. Hence R_N is sparse.

Using the sparsity of R_N, we can reduce the dimension of the problem (2.6.52) to N_R. If we use the iterative methods discussed in § 2.2 to solve (2.6.52), then in each step, we have to compute vector of the form $R_N A_0^+ \xi$ where $\xi \in \mathrm{Im}(R_N)$. From (2.1.16), we see that though ξ is sparse, the computation of $R_N A_0^+ \xi$ still requires $O(n_1^3)$ work and $O(n_1^2)$ storage spaces. These counts are considerably higher than the counts given in (2.6.41) and (2.6.42).

This algorithm has not yet been tested. However, follows from the fast convergence of the capacitance method for elliptic problems and the continuous analogy of the queueing models, we conjecture that this algorithm also has a fast convergence rate.

This concludes our discussion on different algorithms for different overflow models. In the next section, we will analyse theoretically the convergence of our method for the model discussed in § 2.3.1.

Section 3. Analysis of a Model Problem

In this section, we will establish the fast convergence for the 2-queue, single server, one-direction overflow network discussed in § 2.3.1. More precisely, we will obtain an upper bound for the number of iterations required to attain a given accuracy. The main step is to show that the iteration matrix has the form $\omega \cdot I + L + U$, where ω is a scalar, L is a matrix of low rank and U is a matrix of small norm. A good upper bound on the number of iterations can then be obtained by choosing a suitable polynomial in (2.2.3). We note that the convergence theory of the ordinary conjugate gradient method (2.2.1) is much more complete than that of the Orthodir method (2.2.5), see for examples, Axelsson [2] and Van der Vorst [39]. Thus in the following, we will assume that the matrix equations are solved by applying (2.2.1) to the corresponding normal equations. We will first consider the case where the traffic density $\lambda_l < \mu_l$. We will then discuss the cases where $\lambda_l \geq \mu_l$. We note that the problem where $\lambda_l = \mu_l$ is the one we discussed in § 2.2.2, namely, the problem of preconditioning the oblique BVP by the Neumann BVP. We remark that this problem is much more difficult than the problem discussed in § 2.2.1 because the operator is not self-adjoint.

§ 3.1 Defining the Parameters

Recall that when we expand $p_{l,j}$ in (2.3.1) with

$$s_l = 1, \tag{3.1.1}$$

and

$$h_i = (n_i - 1)^{-1}, \qquad i=1,2, \tag{3.1.2}$$

then the underlying continuous equation is of the form (2.3.28). Thus to obtain a sensible limit when the queue sizes n_i, $i=1,2$, are large, we assume that the traffic density and the queue size satisfy the relation

$$\frac{\lambda_i}{\mu_i} = 1 - \beta_i h_i^{\alpha}, \qquad i=1,2, \tag{3.1.3}$$

where μ_i, β_i, $i=1,2$ and α are constants independent of h_i. We remark that

LEMMA 3.1.1

For arbitrary $\{ \lambda_i , \mu_i , h_i \}_{i=1}^{2}$, such that $0 < \lambda_i < \mu_i$ and $0 < h_i \leq 1$, $i=1,2$, then

$$\alpha = \min_{i=1,2} \left\{ \frac{\log (1-\frac{\lambda_i}{\mu_i})}{\log h_i} \right\} > 0, \tag{3.1.4}$$

and (3.1.3) holds with $0 < \beta_i \leq 1$, for $i=1,2$. □

The proof is straightforward. In view of this lemma, we assume in the following that

$$\alpha \geq 0 \text{ and } 0 < \beta_i \leq 1, \quad i=1,2. \tag{3.1.5}$$

Moreover, in order to avoid taking two limits simultaneously, we assume that when h_i, $i=1,2$, tend to zero, the compatibility condition

$$\frac{n_2}{n_1} = C_0 \tag{3.1.6}$$

holds, where C_0 is a constant independent of h_i. We note that by (3.1.3) and (3.1.6)

$$(\frac{\lambda_1}{\mu_1}) / (\frac{\lambda_2}{\mu_2}) = 1 \pm O(h_i^{\alpha}), \tag{3.1.7}$$

or equivalently,

$$\frac{\lambda_1}{\lambda_2} = \frac{\mu_1}{\mu_2} \pm O(h_i^{\alpha}).$$

We assume in the following that there exists a $C > 0$, independent of h_i, such that

$$\frac{1}{C} < \frac{\lambda_1}{\lambda_2} < C. \tag{3.1.8}$$

We remark that we can normalize the problem by setting

$$\mu_1 = 1. \tag{3.1.9}$$

Since $\lambda_1 < \mu_1$, we have

$$\lambda_1 < 1. \tag{3.1.10}$$

In the following, we use C to denote any generic positive constant that depends only on α, μ_i and β_i, $i=1,2$, and is independent of h_i.

To begin the analysis, we first note that by (3.1.1), (2.1.7) becomes

$$S_i = a_i \cdot \mathrm{diag}\,(\,1\,,\,\rho_i\,,\,\ldots\,,\,\rho_i^{n_i-1}\,), \qquad i = 1,2, \tag{3.1.11}$$

where

$$\rho_i = (\frac{\lambda_i}{\mu_i})^{1/4} = (\,1 - \beta_i\, h_i^{\alpha}\,)^{1/4} < 1, \tag{3.1.12}$$

and

$$a_i = \left(\frac{1-\rho_i^2}{1-\rho_i^{2n_i}}\right)^{1/4}. \tag{3.1.13}$$

Notice that for $\alpha \geq 0$ and h_i sufficiently small,

$$\rho_i^{2(n_i-1)} = (\,1 - \beta_i\, h_i^{\alpha}\,)^{\frac{1}{h_i}} = e^{\frac{1}{h_i}\log(\,1\,-\,\beta_i\,h_i^{\alpha}\,)} = e^{-\beta_i h_i^{\alpha-1} - O(h_i^{2\alpha-1})}.$$

Thus we have,

LEMMA 3.1.2

For all $\alpha \geq 0$, there exists $H_i = H_i(\alpha,\beta_i)$, $i = 1,2$, such that for all $h_i < H_i$, we have

$$\rho_i^{2(n_i-1)} \leq e^{-\beta_i h_i^{\alpha-1}} < c_i < 0.5, \qquad \text{if } 0 \leq \alpha < 1, \tag{3.1.14}$$

and

$$\rho_i^{2(n_i-1)} \geq e^{-2\beta_i h_i^{\alpha-1}} > c_i^2 > 0, \qquad \text{if } \alpha \geq 1, \tag{3.1.15}$$

where

$$c_i = c_i(\alpha,\beta_i) = e^{-\beta_i H_i^{\alpha-1}}. \ \square \tag{3.1.16}$$

We remark that when $\alpha \geq 1$ and $h_l < H_l$, by (3.1.15), we also have

$$\rho_l^{(n_l-1)} \geq e^{-\beta_l h_l^{\alpha-1}} \geq 1 - \beta_l h_l^{\alpha-1}. \tag{3.1.17}$$

This follows from the fact that $e^{-x} \geq 1 - x$ when $0 \leq x \leq 1$.

From this lemma, we see that there are three different cases to be considered, namely, the cases where $\alpha < 1$, $\alpha = 1$ and $\alpha > 1$.

§ 3.2 The Case when $\alpha < 1$

For $\alpha < 1$, the problem of solving p in $A\,p = 0$ approaches the separable problem $A_0\,p_0 = 0$. More precisely, we have

THEOREM 3.2.1

If $0 \le \alpha < 1$ and $0 < h_i < H_i$, for $i=1,2$, then

$$\| A\,p_0 \|_2 \le 8\, e^{-\beta_1 h_1^{q-1}}. \tag{3.2.1}$$

Proof: By (2.3.2), (2.1.2), (2.3.5), (2.1.11) and (3.1.11), we have

$$\begin{aligned} A\,p_0 = (A_0 + R_0)\,p_0 &= R_0\,p_0 \\ &= ({}^1e_{n_1}\,{}^1e_{n_1}^* \otimes {}^2R_1)(S_1^2\mathbf{1}_1 \otimes S_2^2\mathbf{1}_2) \\ &= (a_1)^2\,\rho_1^{2(n_1-1)}\,({}^1e_{n_1}\,{}^1e_{n_1}^* \otimes {}^2R_1\,S_2^2\,\mathbf{1}_2). \end{aligned}$$

Thus by (2.3.6), (3.1.10), (3.1.11), (3.1.13) and (3.1.14),

$$\begin{aligned} \| A\,p_0 \|_2 &= (a_1)^2\,\rho_1^{2(n_1-1)}\,\| {}^2R_1\,S_2^2\,\mathbf{1}_2 \|_2 \\ &= \lambda_1\,\rho_1^{2(n_1-1)} \left(\frac{1-\rho_1^2}{1-\rho_1^{2n_1}}\right)\left(\frac{1-\rho_2^2}{1-\rho_2^{2n_2}}\right)\left[2\cdot\frac{1+\rho_2^{4n_2-6}}{(1+\rho_2^2)}\right]^{1/4} \\ &< 2\cdot\frac{1}{1-\rho_1^{2n_1}}\cdot\frac{1}{1-\rho_2^{2n_2}}\,\rho_1^{2(n_1-1)} \\ &< 8\,e^{-\beta_1 h_1^{q-1}}. \quad \square \end{aligned}$$

This lemma shows that when $\alpha < 1$ and n_i sufficiently large, p_0 is already a good approximation to p. In fact, when $\alpha < 1$, $(p_0)_{ij}$ is exponentially small for i close to n_1-1. Hence the direction of the derivative along the boundary $i = n_1-1$, oblique or Neumann, does not have much effect on the solution. As an example, we note that when $\alpha = 0$ and $\rho_i^2 = \beta_i = \frac{1}{4}$, $i=1,2$, then $\| A\,p_0 \|_2 \le 10^{-10}$ when $n_i \ge 32$.

§ 3.3 An Equivalent Problem

In the next few sections, we will analyse the spectrum of the iteration matrix $B^* B$ when $\alpha \geq 1$. Here B is given by (2.3.17). However, to simplify, we will first carry out a similarity transformation of B and work with the transformed matrix. This is the same as if working with B but using a different vector norm. This norm is equivalent to the l_2 norm when $\alpha \geq 1$.

Define

$$B_s = Q_2^* S_2^{-1} B S_2 Q_2 = I + Q_2^* S_2^{-1} {}^2R_1 S_2 Q_2 \Phi. \tag{3.3.1}$$

We note that the transformation matrix is bounded in the l_2 norm. More precisely,

LEMMA 3.3.1

For $\alpha \geq 1$ and $h_2 < H_2$,

$$\| a_2^{-1} S_2 Q_2 \|_2 = a_2^{-1} \| S_2 \|_2 = 1, \tag{3.3.2}$$

and

$$\| a_2 Q_2^* S_2^{-1} \|_2 = a_2 \| S_2^{-1} \|_2 = \rho_2^{-(n_2-1)} \leq \frac{1}{c_2}. \tag{3.3.3}$$

Here a_2 and c_2 are given by (3.1.13) and (3.1.16) respectively. □

The proof follows immediately from (3.1.11) and (3.1.15). We remark that by using (3.1.17), we have a much better bound for (3.3.3), namely,

$$a_2 \| S_2^{-1} \|_2 \leq (1 - \beta_t h_t^{\alpha-1})^{-1}. \tag{3.3.4}$$

Using this lemma, we claim that the spectrum of $B^* B$ and $B_s^* B_s$ are equivalent. More precisely, we have

LEMMA 3.3.2

Let the singular values of B and B_s be $0 \leq \sigma_1 \leq \cdots \leq \sigma_{n_2}$ and $0 \leq \hat{\sigma}_1 \leq \cdots \leq \hat{\sigma}_{n_2}$ respectively. If $\alpha \geq 1$ and $h_2 < H_2$ then

$$c_2 \sigma_j \leq \hat{\sigma}_j \leq \frac{1}{c_2} \sigma_j, \qquad 1 \leq j \leq n_2. \tag{3.3.5}$$

Proof: Given an arbitrary subspace R^j of dimension j in R^{n_2}, it is easy to see that, for all $u \in Q_2^* S_2^{-1} \cdot R^j$,

$$\frac{u^* B_s^* B_s u}{u^* u} = \frac{u^* Q_2^* S_2 B^* S_2^{-2} B S_2 Q_2 u}{u^* u}$$
$$= \frac{z^* a_2^2 S_2^{-2} z}{z^* z} \frac{y^* B^* B y}{y^* y} \frac{x^* a_2^{-2} S_2^2 x}{x^* x},$$

where $x = Q_2 u$, $y = S_2 x$ and $z = B y$. By the Courant-Fischer theorem, see Parlett [34], and lemma 3.3.1,

$$\tilde{\sigma}_j^2 \le \max_{u \in Q_2^* S_2^{-1} R^j} \frac{u^* B_s^* B_s u}{u^* u}$$
$$\le \max_{z \in R^{n_2}} \frac{z^* a_2^2 S_2^{-2} z}{z^* z} \cdot \max_{y \in R^j} \frac{y^* B^* B y}{y^* y} \cdot \max_{x \in R^{n_2}} \frac{x^* a_2^{-2} S_2^2 x}{x^* x}$$
$$\le \frac{1}{c_2^2} \cdot \max_{y \in R^j} \frac{y^* B^* B y}{y^* y} .$$

Hence using the Courant-Fischer theorem again, we have

$$\tilde{\sigma}_j^2 \le \frac{1}{c_2^2} \sigma_j^2.$$

The other inequality in (3.3.5) can be established similarly by using the maximin characterization of the j-th singular value. □

Thus the condition number of $B^* B$ can be expressed in term of the condition number of $B_s^* B_s$. We remark that by using (3.3.4) instead of (3.3.3), we have

$$(1 - \beta_l h_l^{a-1}) \sigma_j \le \tilde{\sigma}_j \le (1 - \beta_l h_l^{a-1})^{-1} \sigma_j, \qquad 1 \le j \le n_2. \tag{3.3.6}$$

Hence for h_l sufficiently small,

$$| \sigma_j - \tilde{\sigma}_j | < 2 \beta_l h_l^{a-1}, \qquad 1 \le j \le n_2. \tag{3.3.7}$$

The tools required to derive the bound for $\tilde{\sigma}_1$ and $\tilde{\sigma}_{n_2}$ are developed in the next section.

§ 3.4 Matrix Identities and the Norm of Φ

In this section, we will prove three lemmas. The first summarizes all the matrix identities we need to derive the bounds on $\tilde{\sigma}_1$ and $\tilde{\sigma}_{n_2}$. The next two give the bounds for the extreme eigenvalues of Φ.

Let us begin by defining

$$R_s = \begin{bmatrix} \rho_2 & & & \\ -1 & \rho_2 & & 0 \\ & \cdot & \cdot & \\ & & \cdot & \cdot \\ & 0 & -1 & \rho_2 \\ & & & -1 \quad 0 \end{bmatrix}, \tag{3.4.1}$$

$$G_s = \begin{bmatrix} \rho_2 & -1 & & & \\ -1 & \rho_2+\frac{1}{\rho_2} & -1 & & 0 \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & 0 & & -1 & \rho_2+\frac{1}{\rho_2} \quad -1 \\ & & & & -1 \quad \frac{1}{\rho_2} \end{bmatrix}, \tag{3.4.2}$$

$$Z_s = \begin{bmatrix} \frac{1}{\rho_2} & 1 & & & \\ -1 & 0 & 1 & & 0 \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & 0 & & -1 & 0 \quad 1 \\ & & & & -1 \quad -\rho_2 \end{bmatrix} \tag{3.4.3}$$

and

$$\Gamma_s = (\lambda_2 \mu_2)^{-\frac{1}{4}} \cdot \Gamma_2 = \mathrm{diag}\,(\gamma_1, \ldots, \gamma_{n_2}) \tag{3.4.4}$$

where

$$\gamma_j = \frac{\gamma_{2,j}}{(\lambda_2 \mu_2)^{\frac{1}{4}}} = \begin{cases} \rho_2 + \dfrac{1}{\rho_2} - 2\cos\theta_j & 1 \le j < n_2, \\ 0 & j = n_2, \end{cases} \tag{3.4.5}$$

with

$$\theta_j = \frac{j\pi}{n_2}, \qquad 1 \le j < n_2. \tag{3.4.6}$$

LEMMA 3.4.1

The following matrix identities hold for all $\rho_2 > 0$.

$$(1) \qquad S_2^{-1} G_2 S_2 = (\lambda_2 \mu_2)^{1/4} G_s, \tag{3.4.7}$$

$$(2) \qquad Q_2^* G_s Q_2 = \Gamma_s, \tag{3.4.8}$$

$$(3) \qquad S_2^{-1\,2} R_1 S_2 = \frac{\lambda_1}{\rho_2} R_s, \tag{3.4.9}$$

$$(4) \qquad R_s R_s^* = \rho_2 G_s, \tag{3.4.10}$$

$$(5) \qquad Z_s G_s + G_s Z_s^* = 0, \tag{3.4.11}$$

$$(6) \qquad Q_2^* Z_s Q_2 \Gamma_s + \Gamma_s Q_2^* Z_s^* Q_2 = 0, \tag{3.4.12}$$

$$(7) \qquad 2 R_s = G_s + (\rho_2 - \frac{1}{\rho_2}) I_{n_2} + Z_s. \tag{3.4.13}$$

Proof (1) and (2) are just restatements of (2.3.20) and (2.1.12) respectively. (3) - (5) can be obtained by straightforward computations. (6) follows from (2) and (5). (7) can be obtained by writing $R_s = \frac{R_s + R_s^*}{2} + \frac{R_s - R_s^*}{2}$. □

We remark that the variables λ_l and μ_l have dimension (1 / time). In contrast, the variables ρ_2 and γ_j, $1 \le j < n_2$, are all dimensionless. Thus the matrices R_s, G_s, Z_s and Γ_s are dimensionless. Accordingly, we define the dimensionless

$$\Phi_s = (\lambda_2 \mu_2)^{1/4} \cdot \Phi, \tag{3.4.14}$$

where Φ is given by (2.3.27). We note that by (3.3.1), (3.4.9) and (3.4.14),

$$B_s = I + \frac{\lambda_1}{\lambda_2} Q_2^* R_s Q_2 \Phi_s. \tag{3.4.15}$$

Moreover, by (2.3.27), (3.4.5) and (3.4.14), the diagonal entries ϕ_j of Φ_s are given by

$$\phi_j = (\lambda_2 \mu_2)^{1/4} \Phi_j = \frac{1}{\gamma_j} \left(1 - \frac{a_j}{\rho_1} \frac{1 - a_j^{2n_1 - 2}}{1 - a_j^{2n_1}} \right), \qquad 1 \le j < n_2, \tag{3.4.16}$$

where a_j are given by (2.3.26). In view of (3.4.5), a_j is the smallest root of

$$a_j^2 - (\rho_1 + \frac{1}{\rho_1} + \zeta\,\gamma_j)\,a_j + 1 = 0, \qquad 1 \le j < n_2, \tag{3.4.17}$$

where

$$\zeta = \left(\frac{\lambda_2\,\mu_2}{\lambda_1\,\mu_1}\right)^{1/4} > 0. \tag{3.4.18}$$

We note that by (3.1.8), there exists a $C > 0$ such that,

$$C^{-1} < \zeta < C. \tag{3.4.19}$$

Since the constant term in (3.4.17) is 1, thus

$$a_j = \frac{\omega_j - (\omega_j^2 - 4)^{1/2}}{2} = \frac{2}{\omega_j + (\omega_j^2 - 4)^{1/2}}, \qquad 1 \le j < n_2, \tag{3.4.20}$$

where

$$\omega_j = \rho_1 + \frac{1}{\rho_1} + \zeta\,\gamma_j, \qquad 1 \le j < n_2. \tag{3.4.21}$$

Since

$$x + \frac{1}{x} \ge 2 \qquad \text{for all } x \ge 0, \tag{3.4.22}$$

$\gamma_j > 0$ by (3.4.5). Hence $\omega_j > 2$ by (3.4.21) and $a_j < 1$ by (3.4.20).

LEMMA 3.4.2

ϕ_j is a decreasing function of j for $1 \le j < n_2$.

Proof: By (3.4.16),

$$\phi_j = \frac{1}{\rho_1}\left(\frac{\rho_1 - a_j}{\gamma_j}\right) + \frac{1}{\rho_1}\left(\frac{1-a_j}{\gamma_j}\right) \cdot \left(a_j^{2n_1-1} \cdot \frac{1+a_j}{1-a_j^{2n_1}}\right), \qquad 1 \le j < n_2. \tag{3.4.23}$$

We claim that the terms in the right hand side are decreasing functions of j. By (3.4.5), we see that γ_j is a increasing function of j for $1 \le j < n_2$. Thus by (3.4.21), ω_j increases w.r.t. j. Hence by (3.4.20), a_j decreases w.r.t. j and so is $a_j^{2n_1-1} \cdot \frac{1 + a_j}{1-a_j^{2n_1}}$. Next we observe that by (3.4.17),

$$(a_j - \rho_1)\,(a_j - \frac{1}{\rho_1}) = \zeta\,\gamma_j\,a_j.$$

Thus the first term in (3.4.23) can be rewritten as

$$\frac{1}{\rho_1}\,\frac{\rho_1-a_j}{\gamma_j} = \frac{\zeta\, a_j}{1-\rho_1 a_j}. \tag{3.4.24}$$

Since a_j, $\rho_1 < 1$, the right hand side is a decreasing function of j. Finally, since

$$\frac{1-a_j}{\gamma_j} = \frac{1-\rho_1}{\gamma_j} + \frac{\rho_1-a_j}{\gamma_j}, \tag{3.4.25}$$

is a sum of two decreasing function of j, $\frac{1}{\rho_1}\left(\frac{1-a_j}{\gamma_j}\right)$ is also a decreasing fuction of j. □

LEMMA 3.4.3

For $\alpha \geq 1$ and $h_i < H_i$, $i = 1, 2$, we have,

(1) $\phi_j \geq C > 0$, for $1 \leq j < n_2$, (3.4.26)

(2) $\phi_j \leq C \cdot \frac{n_2}{j}$, for $1 \leq j < n_2$. (3.4.27)

Proof: To prove (1), we first note that by (3.4.5),

$$\gamma_j = \rho_2 + \frac{1}{\rho_2} - 2\cos\theta_j < \rho_2 + \frac{1}{\rho_2} + 2, \qquad 1 \leq j < n_2. \tag{3.4.28}$$

Hence by (3.4.21)

$$\omega_j < \eta, \qquad 1 \leq j < n_2, \tag{3.4.29}$$

where

$$\eta = \rho_1 + \frac{1}{\rho_1} + \zeta\,(\rho_2 + \frac{1}{\rho_2} + 2). \tag{3.4.30}$$

By (3.4.29), $\omega_j + (\omega_j^2 - 4)^{1/2} < 2\eta$. Thus by (3.4.20),

$$\eta > \frac{\omega_j + (\omega_j^2 - 4)^{1/2}}{2} = \frac{1}{a_j}, \qquad 1 \leq j < n_2. \tag{3.4.31}$$

Hence,

$$\eta - \rho_1 > \frac{1}{a_j} - \rho_1 = \frac{1 - \rho_1 a_j}{a_j}, \qquad 1 \leq j < n_2. \tag{3.4.32}$$

Since a_j, $\rho_1 < 1$, the last term is positive, hence

$$\frac{a_j}{1-\rho_1 a_j} > \frac{1}{\eta-\rho_1} = [\frac{1}{\rho_1} + \zeta\,(\rho_2 + \frac{1}{\rho_2} + 2)]^{-1} > 0. \tag{3.4.33}$$

Notice that the second term in (3.4.23) is nonnegative. Thus by (3.4.24),

$$\phi_j \geq \frac{\rho_1 - a_j}{\rho_1 \gamma_j} = \frac{\zeta a_j}{1-\rho_1 a_j} > \zeta \left[\frac{1}{\rho_1} + \zeta \left(\rho_2 + \frac{1}{\rho_2} + 2\right)\right]^{-1}. \tag{3.4.34}$$

By (3.1.12), $\rho_i = (1 - \beta_i h_i^{\alpha})^{1/\alpha} > (1 - \beta_i H_i^{\alpha})^{1/\alpha} > 0$, for $i = 1,2$, thus

$$\rho_i^{-1} < C, \qquad i = 1,2. \tag{3.4.35}$$

Using this and (3.4.19), (3.4.34) gives $\phi_j \geq C > 0$ for $1 \leq j < n_2$.

Next we prove (2). By (3.4.16),

$$\phi_j = \frac{1}{\gamma_j}\left(1 - \frac{a_j}{\rho_1} \cdot \frac{1-a_j^{2(n_1-1)}}{1-a_j^{2n_1}}\right), \qquad 1 \leq j < n_2.$$

We note that $\dfrac{1-a_j^{2(n_1-1)}}{1-a_j^{2n_1}} \geq \dfrac{n_1-1}{n_1}$. This is because $a_j < 1$ and $f(t) = \dfrac{1}{t}(1 - a^t)$ is a decreasing function of t when $0 < a < 1$ and $t > 0$. Thus

$$\phi_j \leq \frac{1}{\gamma_j}\left(1 - \frac{a_j}{\rho_1}\frac{n_1-1}{n_1}\right) = \frac{\rho_1 - a_j}{\gamma_j \rho_1} + \frac{a_j}{\gamma_j \rho_1 n_1}, \qquad 1 \leq j < n_2. \tag{3.4.36}$$

Notice that in (3.4.24), the right hand side is positive. Thus $\rho_1 \geq a_j$ for $1 \leq j < n_2$. Hence by (3.4.24) again,

$$\phi_j < \frac{\rho_1 - a_j}{\gamma_j \rho_1} + \frac{1}{\gamma_j n_1} = \frac{\zeta a_j}{1-\rho_1 a_j} + \frac{1}{\gamma_j n_1}, \qquad 1 \leq j < n_2. \tag{3.4.37}$$

Since $\rho_1 < 1$,

$$\phi_j < \frac{\zeta a_j}{1-a_j} + \frac{1}{\gamma_j n_1}, \qquad 1 \leq j < n_2. \tag{3.4.38}$$

Notice that by (3.4.17) and (3.4.22), we have

$$(a_j - 1)^2 = \left(\rho_1 + \frac{1}{\rho_1} - 2 + \zeta \gamma_j\right) a_j \geq \zeta \gamma_j a_j, \qquad 1 \leq j < n_2.$$

Since $a_j < 1$, this implies

$$\frac{\zeta a_j}{1-a_j} \leq \left(\frac{\zeta a_j}{\gamma_j}\right)^{1/2}, \qquad 1 \leq j < n_2. \tag{3.4.39}$$

Hence (3.4.38) becomes

$$\phi_j \le (\frac{\zeta\, a_j}{\gamma_j})^{1/2} + \frac{1}{\gamma_j\, n_1}, \qquad 1 \le j < n_2. \tag{3.4.40}$$

Notice that by (3.4.22) and the inequality $\sin\frac{j\pi}{2n_2} \ge \frac{j}{n_2}$, we have

$$\gamma_j = \rho_2 + \frac{1}{\rho_2} - 2 + 4\sin^2\frac{\theta_j}{2} \ge 4\sin^2\frac{\theta_j}{2} \ge \frac{4j^2}{n_2^2}, \qquad 1 \le j < n_2. \tag{3.4.41}$$

Thus by (3.1.6) and (3.4.19), (3.4.40) becomes

$$\phi_j \le \frac{\sqrt{\zeta}\, n_2}{2j} + \frac{n_2^2}{4j^2\, n_1} \le \frac{1}{2}\,(\sqrt{\zeta} + \frac{C_0}{2j})\,\frac{n_2}{j} \le C\cdot\frac{n_2}{j}, \quad 1 \le j < n_2. \quad \square \tag{3.4.42}$$

Recall that by lemmas 2.3.2, ϕ_{n_2} can be defined arbitarily. Thus for simplicity, we assume in the following that

$$\phi_{n_2} = 0. \tag{3.4.43}$$

We remark that this corresponds to a very particular choice of γ in (2.1.15). In fact, by (2.3.18), we see that (3.4.43) holds when

$$\gamma = -\,({}^1q_{n_1,n_1})^{-2} \sum_{j=1}^{n_1-1} ({}^1q_{n_1,j})^2\,(\gamma_{1,j})^{-1}. \tag{3.4.44}$$

Using (3.4.43), lemmas 3.4.2 and 3.4.3, we obtain

$$\| \Phi_j \|_2 \le C\, n_2. \tag{3.4.45}$$

We are now able to derive an upper bound on the condition number of B_j and B.

§ 3.5 Condition Number of B when $\alpha \geq 1$

An upper bound for the largest singular value of B_s can be obtained easily for $\alpha \geq 1$.

LEMMA 3.5.1

For $\alpha \geq 1$ and $h_i < H_i$, $i = 1, 2$,

$$\| B_s \|_2 \leq C \cdot n_2. \tag{3.5.1}$$

Proof By (3.4.15) and (3.4.45)

$$\begin{aligned} \| B_s \|_2 &\leq 1 + \frac{\lambda_1}{\lambda_2} \| Q_2^* R_s Q_2 \Phi_s \|_2 \\ &\leq 1 + \frac{\lambda_1}{\lambda_2} \| R_s \|_2 \| \Phi_s \|_2 \\ &\leq 1 + C \frac{\lambda_1}{\lambda_2} n_2 \| R_s \|_2. \end{aligned}$$

By (3.4.10), (3.4.8) and (3.4.28),

$$\| R_s \|_2^2 = \| R_s R_s^* \|_2 = \rho_2 \| G_s \|_2 = \rho_2 \| \Gamma_s \|_2 \leq (\rho_2 + 1)^2 \leq 4.$$

Thus $\| B_s \|_2 \leq 1 + C \frac{\lambda_1}{\lambda_2} n_2$. By (3.1.8), the lemma follows. □

To derive a lower bound for the smallest singular value of $B_s^* B_s$, we need

LEMMA 3.5.2

Let $B_w = B_s W$ where W is nonsingular. If $\lambda_{\min}(B_w + B_w^*) \geq \delta > 0$, then

$$\| B_s^{-1} \|_2 \leq \frac{2}{\delta} \| W \|_2. \tag{3.5.2}$$

Proof For arbitrary x, using the Cauchy-Schwartz inequality

$$\delta \| x \|_2^2 \leq \lambda_{\min}(B_w + B_w^*) \| x \|_2^2 \leq x^* (B_w + B_w^*) x = 2 x^* B_w x \leq 2 \| x \|_2 \| B_w x \|_2.$$

Define $y = B_w x$, we have,

$$\frac{\| B_w^{-1} y \|_2}{\| y \|_2} \leq \frac{2}{\delta}.$$

Since y is arbitrary, hence $\| B_w^{-1} \|_2 \le \frac{2}{\delta}$ and

$$\| B_s^{-1} \|_2 = \| W W^{-1} B_s^{-1} \|_2 \le \| W \|_2 \| B_w^{-1} \|_2 \le \frac{2}{\delta} \| W \|_2. \quad \square$$

To find an appropriate W, we first define the generalized inverse Φ_s^+ of Φ_s as

$$\Phi_s^+ = \text{diag} (\phi_1^{-1}, \ldots, \phi_{n_2-1}^{-1}, 1). \tag{3.5.3}$$

Define also the extension Γ_0 of Γ_s as

$$\Gamma_0 = \text{diag} (\gamma_1, \ldots, \gamma_{n_2-1}, 1). \tag{3.5.4}$$

We note that Φ_s^+ and Γ_0 are nonsingular, and by (3.4.43)

$$\Phi_s \Phi_s^+ \Gamma_0 = \Gamma_s. \tag{3.5.5}$$

Let us define B_0 by

$$B_0 = B_s \Phi_s^+ \Gamma_0. \tag{3.5.6}$$

We establish

LEMMA 3.5.3

For all $\alpha > 1$, there exists an $0 < H_3 \le \min \{ H_1, H_2 \}$, such that for all $h_1, h_2 < H_3$, we have

$$\lambda_{\min}(B_0 + B_0^*) \ge C \cdot h_2^3. \tag{3.5.7}$$

Proof: By (3.4.15), (3.5.5), (3.4.13) and (3.4.8),

$$\begin{aligned} B_0 &= (I + \frac{\lambda_1}{\lambda_2} Q_2^* R_s Q_2 \Phi_s) \Phi_s^+ \Gamma_0 \\ &= \Phi_s^+ \Gamma_0 + \frac{\lambda_1}{\lambda_2} Q_2^* R_s Q_2 \Gamma_s \\ &= \Phi_s^+ \Gamma_0 + \frac{1}{2} \frac{\lambda_1}{\lambda_2} [\Gamma_s + (\rho_2 - \frac{1}{\rho_2}) \cdot I + Q_2^* Z_2 Q_2] \Gamma_s. \end{aligned}$$

Thus by (3.4.12),

$$B_0^* + B_0 = 2\Phi_s^+ \Gamma_0 + \frac{\lambda_1}{\lambda_2} [\Gamma_s + (\rho_2 - \frac{1}{\rho_2}) I] \Gamma_s,$$

which is a diagonal matrix. Hence by (3.5.3), (3.5.4) and (3.4.5)

$$\lambda_{\min}(B_0^* + B_0) = \min_{1 \le j < n_2} \{ \gamma_j \cdot [2\phi_j^{-1} + \frac{\lambda_1}{\lambda_2} (\gamma_j + \rho_2 - \frac{1}{\rho_2})] , 2 \}$$

$$= 2 \cdot \min_{1 \le j < n_2} \{ \gamma_j \cdot [\phi_j^{-1} + \frac{\lambda_1}{\lambda_2} (\rho_2 - \cos\theta_j)] , 1 \}.$$

By lemma 3.4.2, ϕ_j^{-1} is an increasing function of j for $1 \le j < n_2$, and by (3.4.5) and (3.4.6), so are γ_j and $-\cos\theta_j$. Thus

$$\lambda_{\min}(B_0^* + B_0) \ge 2 \cdot \min \{ \gamma_1 [\phi_1^{-1} + \frac{\lambda_1}{\lambda_2} (\rho_2 - \cos\theta_1)] , 1 \}. \tag{3.5.8}$$

Notice that by lemma 3.4.3, $\phi_1^{-1} \ge C \cdot h_2$, while by (3.1.3) and (3.4.6),

$$\rho_2 - \cos\theta_1 = \pi^2 h_2^2 - \frac{1}{2} \beta_2 h_2^{\alpha} + \text{higher order terms.} \tag{3.5.9}$$

Thus when $\alpha > 1$ and h_2 sufficiently small, we have $\phi_1^{-1} + \frac{\lambda_1}{\lambda_2} (\rho_2 - \cos\theta_1) > C \cdot h_2 > 0$. This implies that $\lambda_{\min}(B_0^* + B_0) \ge C \cdot \gamma_1 h_2$. Hence by (3.4.41), we get (3.5.7).

For $\alpha = 1$, by (3.4.42) and (3.5.9), we see that for all sufficiently small h_2, the right hand side in (3.5.8) is positive if

$$2 (\sqrt{\zeta} + \frac{C_0}{2})^{-1} > \frac{1}{2} \beta_2 \frac{\lambda_1}{\lambda_2}. \tag{3.5.10}$$

Thus we have

Corollary 3.5.4

If $\alpha = 1$ and (3.5.10) holds, then there exists an $0 < H_3 \le \min \{ H_1, H_2 \}$, such that for all $h_1, h_2 < H_3$, (3.5.7) holds. □

We remark that if $\lambda_1 = \lambda_2$, $\mu_1 = \mu_2$ and $h_1 = h_2$, then (3.5.10) holds for all $0 \le \beta_2 \le 1$.

By (3.5.3) and (3.4.34),

$$\| \Phi_s^+ \|_2 \le \max_{1 \le j < n_2} \{ \phi_j^{-1} , 1 \} \le \{ \rho_2 + \frac{1}{\rho_2} + 2 + \frac{1}{\zeta \rho_1} \} < C.$$

Moreover, by (3.5.4) and (3.4.28),

$$\| \Gamma_0 \|_2 \le \rho_2 + \frac{1}{\rho_2} + 2 < C.$$

Thus by (3.4.19) and (3.4.35),

$$\| \Phi_s^+ \Gamma_0 \|_2 < C. \tag{3.5.11}$$

Hence by (3.5.6), (3.5.7) and lemma 3.5.2, we get

LEMMA 3.5.5

Assume that either $\alpha = 1$ and (3.5.10) holds or $\alpha > 1$, then for $h_1, h_2 < H_3$, we have

$$\| B_s^{-1} \|_2 \leq C \cdot n_2^3. \quad \square \tag{3.5.12}$$

Combining this result with lemma 3.5.1, we obtain

THEOREM 3.5.6

Let $\kappa(B_s)$ and $\kappa(B)$ be the condition number of B_s and B respectively. Assume that either $\alpha = 1$ and (3.5.10) holds or $\alpha > 1$, then for $h_1, h_2 < H_3$, we have

$$\kappa(B_s) \leq C \cdot n_2^4 \tag{3.5.13}$$

and

$$\kappa(B) \leq C \cdot n_2^4. \quad \square \tag{3.5.14}$$

This theorem and (2.2.4) suggest that the convergence rate of the ordinary conjugate gradient method (2.2.1), when applied to the normal equations corresponding to B or B_s, may be extremely slow. However, in § 3.8, we will show that the method converges quickly as a consequence of a clustering of the singular values. To do so, we need more information about the matrix Φ_s and Z_s. For simplicity, we will also reduce the dimension of the problem further to $n_2 - 1$.

§ 3.6 More Matrix Identities and the Approximation of Φ

In this section, we prove three lemmas that are the keys in proving, in the next section, that the singular values of B are clustered. The first two lemmas summarize all the matrix identities we will be using. The last one gives an asymptotic approximation of the matrix Φ_s when n_l are large.

Define the projection Q_p from Q_2 as the n_2 by n_2-1 matrix

$$Q_p = [\, q_1 , \ldots , q_{n_2-1} \,], \tag{3.6.1}$$

where q_l are the i-th column of Q_2. Define also the n_2-1 by n_2-1 matrices

$$\Phi_p = \mathrm{diag}\,(\, \phi_1 , \ldots , \phi_{n_2-1} \,), \tag{3.6.2}$$

$$\Gamma_p = \mathrm{diag}\,(\, \gamma_1 , \ldots , \gamma_{n_2-1} \,), \tag{3.6.3}$$

$$L_1 = \frac{1}{\rho_2}\, Q_p^*\, {}^2e_1\, {}^2e_1^*\, Q_p, \tag{3.6.4}$$

$$L_2 = \rho_2\, Q_p^*\, {}^2e_{n_2}\, {}^2e_{n_2}^*\, Q_p, \tag{3.6.5}$$

$$L_0 = L_1 - L_2, \tag{3.6.6}$$

$$L_3 = \begin{bmatrix} 1 & 0 & & & \\ 0 & 0 & 0 & & 0 \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & 0 \\ & 0 & & 0 & 0 & \rho_2 \\ & & & & \rho_2 & -1 \end{bmatrix} \tag{3.6.7}$$

and

$$L_4 = \Phi_p\, L_0 + L_0\, \Phi_p. \tag{3.6.8}$$

We remark that L_1 and L_2 are matrices of rank 1, while L_0, L_3 and L_4 are matrices of rank 2, 3 and 4 respectively. Moreover, these matrices are symmetric. We note that by (2.3.23) and (2.3.21),

$$(\, L_1 \,)_{kj} = \frac{1}{\rho_2}\,\frac{2}{n_2} \sin \psi_{2k} \sin \psi_{2j} = (-1)^{k+j}\, (\, L_2 \,)_{kj}, \qquad 1 \le k\,, j < n_2. \tag{3.6.9}$$

LEMMA 3.6.1

The following identities hold for all $\rho_2 > 0$.

$$(1) \qquad R_s^* R_s = \rho_2 G_s + L_3, \tag{3.6.10}$$

$$(2) \qquad L_0 = \frac{1}{2} Q_p^* (Z_s^* + Z_s) Q_p, \tag{3.6.11}$$

$$(3) \qquad R_s^* Q_2 \cdot {}^2e_{n_2} = \mathbf{0}_{n_2}, \tag{3.6.12}$$

$$(4) \qquad Q_2^* Z_s Q_2 \Phi_s = \begin{bmatrix} Q_p^* Z_s Q_p \Phi_p & \mathbf{0}_{n_2-1} \\ \mathbf{0}_{n_2-1}^* & 0 \end{bmatrix}, \tag{3.6.13}$$

$$(5) \qquad Q_2^* Z_s Q_2 \Gamma_s = \begin{bmatrix} Q_p^* Z_s Q_p \Gamma_p & \mathbf{0}_{n_2-1} \\ \mathbf{0}_{n_2-1}^* & 0 \end{bmatrix}, \tag{3.6.14}$$

$$(6) \qquad \frac{1}{2}\{ Q_p^* Z_s Q_p \Phi_p + \Phi_p Q_p^* Z_s^* Q_p \}$$

$$= L_4 - \frac{1}{2}\{ \Phi_p Q_p^* Z_s Q_p + Q_p^* Z_s^* Q_p \Phi_p \}. \tag{3.6.15}$$

Proof: (1) and (2) can be shown by straightforward computations. To prove (3), we note that by (2.3.23), $Q_2 {}^2e_{n_2} = S_2 \mathbf{1}_2 = a_2 (1 , \rho_2 , \ldots , \rho_2^{n_2-1})^*$, where a_2 is given by (3.1.13). Thus by (3.4.1), (3) follows. To prove (4), observe that by (3.4.43), $Q_2^* Z_s Q_2 \Phi_s {}^2e_{n_2} = \mathbf{0}_{n_2}$. Notice also that by (3.4.13), (3.4.8), (3.6.12) and (3.4.43)

$$\begin{aligned} {}^2e_{n_2}^* Q_2^* Z_s Q_2 \Phi_s &= \{ 2 \cdot {}^2e_{n_2}^* Q_2^* R_s Q_2 - {}^2e_{n_2}^* (\rho_2 - \frac{1}{\rho_2}) \cdot I - {}^2e_{n_2}^* \Gamma_s \} \Phi_s \\ &= (\frac{1}{\rho_2} - \rho_2) \cdot {}^2e_{n_2}^* \Phi_s = \mathbf{0}_{n_2}. \end{aligned}$$

The proof of (5) is similar to that of (4). (6) follows from (2) and (3.6.8). □

For $i = 1,2$, define the (n_2-1) by (n_2-1) matrices

$$W_l = \sum_{l=1}^{n_1-1} \tilde{\gamma}_l ({}^1q_{n_1,l})^2 (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} L_l (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1}, \tag{3.6.16}$$

where $({}^1q_{n_1,l})$, given by (2.3.23), is the (n_1,l) entry of Q_1 and

$$\tilde{\gamma}_l = \frac{\gamma_{1,l}}{(\lambda_2 \mu_2)^{\frac{1}{4}}} = \begin{cases} \frac{1}{\zeta} (\rho_1 + \frac{1}{\rho_1} - 2 \cos \frac{l\pi}{n_1}) & 1 \le l < n_1, \\ 0 & l = n_1. \end{cases} \tag{3.6.17}$$

Here $\gamma_{1,j}$ and ζ given by (2.3.24) and (3.4.18) respectively. We have,

LEMMA 3.6.2

W_1 and W_2 are positive semi-definite matrices and satisfy

$$(1) \qquad (W_1)_{kj} = (-1)^{k+j} (W_2)_{kj}, \qquad 1 \le k, j \le n_2, \tag{3.6.18}$$

$$(2) \qquad W_1 - W_2 = \frac{1}{2}\{ \Phi_p Q_p^* Z_s Q_p + Q_p^* Z_s^* Q_p \Phi_p^* \}. \tag{3.6.19}$$

Proof: To prove that the W_l are positive semi-definite, we first note that, the L_l are symmetric, and the $(\Gamma_p + \tilde{\gamma}_l \cdot I)$, $1 \le l < n_1$, are diagonal. Thus the W_l are symmetric. Next we observe that by (3.6.17) and (3.4.22),

$$\tilde{\gamma}_l \ge \frac{1}{\zeta}\left(2 - 2\cos\frac{l\pi}{n_1}\right) = \frac{4}{\zeta}\sin^2\frac{l\pi}{n_1} > 0, \qquad 1 \le l < n_1. \tag{3.6.20}$$

Hence for all $x \in R^{n_2-1}$, by (3.6.16), (3.6.4) and (3.6.5),

$$x^* W_1 x = \frac{1}{\rho_2}\sum_{l=1}^{n_1-1} \tilde{\gamma}_l \left({}^1q_{n_1,l}\right)^2 \left[x^* (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} Q_p^{*2} e_1\right]^2 \ge 0, \tag{3.6.21}$$

$$x^* W_2 x = \rho_2 \sum_{l=1}^{n_1-1} \tilde{\gamma}_l \left({}^1q_{n_1,l}\right)^2 \left[x^* (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} Q_p^{*2} e_{n_2}\right]^2 \ge 0. \tag{3.6.22}$$

Thus the W_l are positive semi-definite.

To prove (1), we observe that by (3.6.16),

$$(W_l)_{kj} = (L_l)_{kj} \sum_{l=1}^{n_1-1} \frac{\tilde{\gamma}_l \left({}^1q_{n_1,l}\right)^2}{(\gamma_k + \tilde{\gamma}_l)(\gamma_j + \tilde{\gamma}_l)}, \quad 1 \le k, j < n_2, \quad i = 1, 2. \tag{3.6.23}$$

Thus by (3.6.9), (1) follows.

Finally we prove (2). By (3.6.13) and (3.4.12),

$$Q_p^* Z_s Q_p \Gamma_p + \Gamma_p Q_p^* Z_s^* Q_p = 0. \tag{3.6.24}$$

Thus for $1 \le l \le n_1$, by (3.6.11), we have

$$Q_p^* Z_s Q_p (\Gamma_p + \tilde{\gamma}_l \cdot I) + (\Gamma_p + \tilde{\gamma}_l \cdot I) Q_p^* Z_s^* Q_p = \tilde{\gamma}_l \{ Q_p^* (Z_s + Z_s^*) Q_p \}$$
$$= 2\tilde{\gamma}_l L_0. \tag{3.6.25}$$

Notice that by (3.4.41), $\gamma_j > 0$ for $1 \le j < n_2$. Thus Γ_p is positive definite. By (3.6.17) and (3.6.20), we see that $\tilde{\gamma}_l \ge 0$ for $1 \le l \le n_1$. Hence $(\Gamma_p + \tilde{\gamma}_l \cdot I)$, $1 \le l \le n_1$, are positive

definite and hence invertible. Thus (3.6.25) gives

$$\begin{aligned}&(\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} Q_p^* Z_s Q_p + Q_p^* Z_s^* Q_p (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} \\ &= 2\,\tilde{\gamma}_l (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} L_0 (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1}, \qquad 1 \le l \le n_1. \end{aligned} \tag{3.6.26}$$

Recall that by (2.3.16), (2.1.14) and (2.1.15),

$$\Phi = \sum_{l=1}^{n_1-1} ({}^1q_{n_1,l})^2 \cdot (\Gamma_2 + \gamma_{1,l} \cdot I)^{-1} + ({}^1q_{n_1,n_1})^2 \Gamma_2^+.$$

Thus by (3.4.14), (3.4.4) and (3.6.17),

$$\Phi_s = \sum_{l=1}^{n_1-1} ({}^1q_{n_1,l})^2 \cdot (\Gamma_s + \tilde{\gamma}_l \cdot I)^{-1} + ({}^1q_{n_1,n_1})^2 \Gamma_s^+.$$

Restricting to the (n_2-1)-dimensional space, we have, by (3.6.2) and (3.6.3),

$$\Phi_p = \sum_{l=1}^{n_1-1} ({}^1q_{n_1,l})^2 \cdot (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} + ({}^1q_{n_1,n_1})^2 \Gamma_p^{-1}.$$

Since $\tilde{\gamma}_{n_1} = 0$,

$$\Phi_p = \sum_{l=1}^{n_1} ({}^1q_{n_1,l})^2 \cdot (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1}. \tag{3.6.27}$$

Thus multiplying (3.6.26) by $({}^1q_{n_1,l})^2$ and taking the sum from $l=1$ to n_1, we get

$$\begin{aligned}\Phi_p Q_p^* Z_s Q_p + Q_p^* Z_s^* Q_p \Phi_p^* &= 2 \sum_{l=1}^{n_1-1} \tilde{\gamma}_l ({}^1q_{n_1,l})^2 \cdot (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} L_0 (\Gamma_p + \tilde{\gamma}_l \cdot I)^{-1} \\ &= 2\{W_1 - W_2\},\end{aligned}$$

where the last equality follows from (3.6.6) and (3.6.16). □

Finally we give an approximation $\tilde{\Phi}_p$ of Φ_p. According to (3.4.23), we define

$$\tilde{\Phi}_p = \operatorname{diag}(\tilde{\phi}_1, \ldots, \tilde{\phi}_{n_2-1}), \tag{3.6.28}$$

where $\tilde{\phi}_j$ is the first term of (3.4.23), i.e.,

$$\tilde{\phi}_j = \frac{\rho_1 - a_j}{\rho_1 \gamma_j}, \qquad 1 \le j < n_2. \tag{3.6.29}$$

Define also the constant

$$\zeta_0 = \frac{C_0}{\sqrt{\zeta}} = \frac{n_2}{n_1}\left(\frac{\lambda_1\mu_1}{\lambda_2\mu_2}\right)^{\frac{1}{4}}, \tag{3.6.30}$$

where C_0 is given by (3.1.6). We remark that ζ_0 depends on h_i, $i = 1$ and 2. However, by (3.1.8), there exists a C_1, independent of h_1 and h_2, such that

$$\zeta_0 < C_1. \tag{3.6.31}$$

LEMMA 3.6.3

For all $\alpha \geq 1$, there exists an $0 < H_4 < H_3$ and $C > 0$ such that for h_1 and $h_2 < H_4$ and $C_1 \log n_2 \leq j < n_2$, we have

(1) $$a_j^{2n_1} < C / n_2^3, \tag{3.6.32}$$

(2) $$0 < \phi_j - \tilde{\phi}_j < \frac{C}{j^2 n_2}. \tag{3.6.33}$$

Proof: We prove (1) first. Observe that by (3.4.21), (3.4.22) and (3.4.41),

$$\omega_j > 2 + \zeta \gamma_j > 2 + 4\zeta \frac{j^2}{n_2^2}, \qquad 1 \leq j < n_2.$$

Hence

$$(\omega_j^2 - 4)^{\frac{1}{4}} \geq 4\sqrt{\zeta}\, j / n_2, \qquad 1 \leq j < n_2.$$

Thus (3.4.20) gives

$$a_j < 2\left(2 + 4\zeta \frac{j^2}{n_2^2} + 4\sqrt{\zeta}\frac{j}{n_2}\right)^{-1} < \left(1 + \sqrt{\zeta}\frac{j}{n_2}\right)^{-2}, \quad 1 \leq j < n_2. \tag{3.6.34}$$

If $j \geq \frac{n_2}{\sqrt{\zeta}}$, then (3.6.34) and (3.1.6) give

$$a_j^{2n_1} \leq (1+1)^{-4n_1} = e^{-4n_1 \log 2} \leq e^{-2n_1} \leq \frac{1}{n_1^3} \leq \frac{C_0^3}{n_2^3}. \tag{3.6.35}$$

If $j < \frac{n_2}{\sqrt{\zeta}}$, then (3.6.34) and (3.1.6) give

$$a_j^{2n_1} \leq \left(1 + \sqrt{\zeta}\frac{j}{n_2}\right)^{-4n_1} = e^{-4n_1 \log\left(1+\sqrt{\zeta}\frac{j}{n_2}\right)} = e^{-4\sqrt{\zeta} j / C_0 + O(h_2)} = e^{-4j/\zeta_0 + O(h_2)}.$$

By (3.6.31), we have

$$a_j^{2n_1} \le e^{-4j/C_1 + O(h_2)}, \qquad \text{for } 1 \le j < \frac{n_2}{\sqrt{\zeta}}.$$

Thus for h_1 and h_2 sufficiently small,

$$a_j^{2n_1} \le e^{-3j/C_1}, \qquad \text{for } 1 \le j < \frac{n_2}{\sqrt{\zeta}}. \tag{3.6.36}$$

Hence if $\frac{n_2}{\sqrt{\zeta}} > j \ge C_1 \log n_2$,

$$a_j^{2n_1} \le e^{-3\frac{C_1}{C_1}\log n_2} \le e^{-3\log n_2} \le \frac{1}{n_2^3}. \tag{3.6.37}$$

Combining this with (3.6.35), we get (1).

To prove (2), we observe that by (3.4.23) and the definition of $\tilde{\phi}_j$, we have

$$\phi_j - \tilde{\phi}_j = \frac{1 - a_j^2}{\rho_1\ a_j\,(1 - a_j^{2n_1})}\,\frac{a_j^{2n_1}}{\gamma_j}, \qquad 1 \le j < n_2. \tag{3.6.38}$$

Since $a_j < 1$, this implies $\phi_j - \tilde{\phi}_j > 0$. By (3.4.31), (3.4.30), (3.4.19) and (3.4.35),

$$\frac{1}{a_j} \le \rho_1 + \frac{1}{\rho_1} + \zeta\,(\,\rho_2 + \frac{1}{\rho_2} + 2\,) < C, \qquad 1 \le j < n_2. \tag{3.6.39}$$

Thus from (3.6.32), we see that for $h_1 < H_4$ and $C_1 \log n_2 \le j < n_2$,

$$\frac{1 - a_j^2}{\rho_1\, a_j\,(1 - a_j^{2n_1})} \le \frac{1 + C\,h_2^3}{\rho_1\, a_j} \le C, \qquad \text{for } C_1 \log n_2 \le j < n_2.$$

Hence by (3.6.32) and (3.4.41), (3.6.38) becomes

$$\phi_j - \tilde{\phi}_j \le \frac{C\,a_j^{2n_1}}{\gamma_j} \le \frac{C}{\gamma_j\, n_2^3} \le \frac{C}{j^2\, n_2}, \qquad \text{for } C_1 \log n_2 \le j < n_2. \quad \square$$

§ 3.7 Clustering of Singular Values of B

Using the lemmas in the previous section, we are able to prove that the singular values of B are clustered. More precisely, we will show that $B^* B = (1 + \frac{\lambda_1}{\lambda_2}) \cdot I + L + U$, where L is a matrix of low rank and U is a matrix of small l_2 norm.

Recalling (3.4.15) and using (3.6.10), we have

$$B_j^* B_j = I + \frac{\lambda_1}{\lambda_2} \{ Q_2^* R_j Q_2 \Phi_j + \Phi_j Q_2^* R_j^* \Phi_j \} + \rho_2 (\frac{\lambda_1}{\lambda_2})^2 \Phi_j Q_2^* G_j Q_2 \Phi_j + (\frac{\lambda_1}{\lambda_2})^2 \Phi_j Q_2^* L_3 Q_2 \Phi_j. \tag{3.7.1}$$

For simplicity, we will use $\hat{L}_j$ to denote matrices of rank less than or equal to j. Using (3.4.13) and (3.4.8), (3.7.1) becomes

$$\begin{aligned} B_j^* B_j &= I + \frac{\lambda_1}{\lambda_2} \{ \Gamma_j + (\rho_2 - \frac{1}{\rho_2}) I + \rho_2 (\frac{\lambda_1}{\lambda_2}) \Phi_j \Gamma_j \} \Phi_j \\ &\quad + \frac{1}{2} (\frac{\lambda_1}{\lambda_2}) \{ Q_2^* Z_j Q_2 \Phi_j + \Phi_j Q_2^* Z_j^* \Phi_j \} + \hat{L}_3 \\ &= D + W + \hat{L}_3, \end{aligned} \tag{3.7.2}$$

where

$$D = I + \frac{\lambda_1}{\lambda_2} \{ \Gamma_j + (\rho_2 - \frac{1}{\rho_2}) I + \rho_2 (\frac{\lambda_1}{\lambda_2}) \Phi_j \Gamma_j \} \Phi_j \tag{3.7.3}$$

is a diagonal matrix, and

$$W = \frac{1}{2} (\frac{\lambda_1}{\lambda_2}) \{ Q_2^* Z_j Q_2 \Phi_j + \Phi_j Q_2^* Z_j^* \Phi_j \}. \tag{3.7.4}$$

By (3.6.13), $W = \begin{bmatrix} W_p & \mathbf{0}_{n_2-1} \\ \mathbf{0}_{n_2-1}^* & 0 \end{bmatrix}$, where

$$W_p = \frac{1}{2} (\frac{\lambda_1}{\lambda_2}) \{ Q_p^* Z_j Q_p \Phi_p + \Phi_p Q_p^* Z_j^* \Phi_p \}$$

is a matrix of order $n_2 - 1$. By (3.6.15) and (3.6.19),

$$\begin{aligned} W_p &= \frac{\lambda_1}{\lambda_2} \{ L_4 - \frac{1}{2} (\Phi_p Q_p^* Z_j Q_p + Q_p^* Z_j^* Q_p \Phi_p) \} \\ &= \frac{\lambda_1}{\lambda_2} \{ L_4 + W_2 - W_1 \}. \end{aligned} \tag{3.7.5}$$

By (3.4.43), $D = \begin{bmatrix} D_p & 0_{n_2-1} \\ 0^*_{n_2-1} & 1 \end{bmatrix}$. Here D_p is a diagonal matrix of order n_2-1 and is given by

$$D_p = \text{diag}\,(d_1, \ldots, d_{n_2-1})$$
$$= I + \frac{\lambda_1}{\lambda_2}\{\Gamma_p + (\rho_2 - \frac{1}{\rho_2})I + \rho_2(\frac{\lambda_1}{\lambda_2})\Phi_p\Gamma_p\}\Phi_p. \tag{3.7.6}$$

Combining this with (3.7.5), (3.7.2) gives

$$B_s^* B_s = \begin{bmatrix} D_p + W_p & 0_{n_2-1} \\ 0^*_{n_2-1} & 1 \end{bmatrix} + \tilde{L}_3$$
$$= \begin{bmatrix} D_p + \frac{\lambda_1}{\lambda_2}(W_2 - W_1) & 0_{n_2-1} \\ 0^*_{n_2-1} & 1 + \frac{\lambda_1}{\lambda_2} \end{bmatrix} + \tilde{L}_8. \tag{3.7.7}$$

Corresponding to (3.6.28), we define the approximation $\tilde{D}_p$ of D_p as

$$\tilde{D}_p = \text{diag}\,(\tilde{d}_1, \ldots, \tilde{d}_{n_2-1})$$
$$= I + \frac{\lambda_1}{\lambda_2}\{\Gamma_p + (\rho_2 - \frac{1}{\rho_2})I + \rho_2(\frac{\lambda_1}{\lambda_2})\tilde{\Phi}_p\Gamma_p\}\tilde{\Phi}_p. \tag{3.7.8}$$

LEMMA 3.7.1

For $\alpha \geq 1$, there exists a $C_2 > 0$ such that for $h_i < H_4$, $i = 1,2$,

$$|\tilde{d}_j - d_j| < \frac{C_2 h_2}{j^2}, \quad \text{for } C_1 \log n_2 \leq j < n_2. \tag{3.7.9}$$

Here C_1 and H_4 are given by (3.6.31) and lemma 3.6.3 respectively.

Proof: By (3.7.6) and (3.7.8),

$$d_j - \tilde{d}_j = (\frac{\lambda_1}{\lambda_2})\{\gamma_j + (\rho_2 - \frac{1}{\rho_2})I + \rho_2(\frac{\lambda_1}{\lambda_2})\gamma_j\phi_j\}(\phi_j - \tilde{\phi}_j)$$
$$+ (\frac{\lambda_1}{\lambda_2})^2\rho_2\gamma_j\tilde{\phi}_j(\phi_j - \tilde{\phi}_j), \qquad 1 \leq j < n_2. \tag{3.7.10}$$

Notice that by (3.4.40) and the fact that $a_j < 1$, we have

$$\gamma_j\phi_j \leq (\zeta\gamma_j)^* + n_1^{-1}, \qquad 1 \leq j < n_2. \tag{3.7.11}$$

Thus by (3.4.28), (3.4.19) and (3.4.35),

$$\gamma_j \phi_j \le C, \qquad 1 \le j < n_2. \tag{3.7.12}$$

Hence by (3.6.33),

$$\gamma_j \tilde{\phi}_j \le \gamma_j \phi_j \le C, \qquad C_1 \log n_2 \le j < n_2. \tag{3.7.13}$$

Thus by (3.1.8), (3.4.28) and (3.6.33), (3.7.10) gives

$$| d_j - \tilde{d}_j | < C | \phi_j - \tilde{\phi}_j | < \frac{C_2 h_2}{j^2}, \qquad C_1 \log n_2 \le j < n_2. \ \square$$

LEMMA 3.7.2

For $\alpha \ge 1$, there exists a constant $C_3 > 0$ such that when $h_i < H_4$, $i = 1,2$,

$$| \tilde{d}_j - (1 + \frac{\lambda_1}{\lambda_2}) | < C_3 h_2^{\alpha-1} / j \qquad 1 \le j < n_2. \tag{3.7.14}$$

Proof: By (3.7.8) and (3.6.29), we have

$$\tilde{d}_j = 1 + \frac{\lambda_1}{\lambda_2} \frac{\rho_1 - a_j}{\rho_1 \gamma_j} \{ \gamma_j + (\rho_2 - \frac{1}{\rho_2}) + \rho_2 (\frac{\lambda_1}{\lambda_2}) \frac{\rho_1 - a_j}{\rho_1} \}, \qquad 1 \le j < n_2.$$

By definition (3.4.18), $\zeta = \frac{\rho_1 \lambda_2}{\rho_2 \lambda_1}$. Thus

$$\begin{aligned} \tilde{d}_j &= 1 + \frac{\lambda_1}{\lambda_2} \frac{\rho_1 - a_j}{\rho_1 \gamma_j} \{ \gamma_j + (\rho_2 - \frac{1}{\rho_2}) + \frac{\rho_1 - a_j}{\zeta} \} \\ &= 1 + \frac{\lambda_1}{\lambda_2} + \frac{\lambda_1}{\lambda_2} \frac{(\rho_1 - a_j)}{\rho_1 \gamma_j \zeta} \{ - \frac{\zeta a_j \gamma_j}{\rho_1 - a_j} + \zeta (\rho_2 - \frac{1}{\rho_2}) + (\rho_1 - a_j) \}. \end{aligned}$$

By (3.4.24), this can be rewritten as

$$\begin{aligned} (1 + \frac{\lambda_1}{\lambda_2}) - \tilde{d}_j &= \frac{\lambda_1}{\lambda_2} \frac{a_j}{1 - \rho_1 a_j} \{ \frac{1 - \rho_1 a_j}{\rho_1} + \zeta (\frac{1}{\rho_2} - \rho_2) - \rho_1 + a_j \} \\ &= \frac{\lambda_1}{\lambda_2} \frac{a_j}{1 - \rho_1 a_j} \{ (\frac{1}{\rho_1} - \rho_1) + \zeta (\frac{1}{\rho_2} - \rho_2) \}. \end{aligned} \tag{3.7.15}$$

Since $a_j, \rho_1 < 1$,

$$0 \le \frac{a_j}{1 - \rho_1 a_j} \le \frac{a_j}{1 - a_j}, \qquad 1 \le j < n_2. \tag{3.7.16}$$

Hence we have

$$| \tilde{d}_j - (1 + \frac{\lambda_1}{\lambda_2}) | \le \frac{\lambda_1}{\lambda_2} \frac{a_j}{1 - a_j} | (\frac{1}{\rho_1} - \rho_1) + \zeta (\frac{1}{\rho_2} - \rho_2) |, \ 1 \le j < n_2. \tag{3.7.17}$$

Notice that by (3.4.39) and (3.4.41),

$$\frac{a_j}{1-a_j} \le \left(\frac{a_j}{\zeta\,\gamma_j}\right)^{\frac{1}{2}} \le \frac{1}{\sqrt{\zeta\,\gamma_j}} \le \frac{n_2}{2\sqrt{\zeta}\,j}, \qquad 1 \le j < n_2. \tag{3.7.18}$$

On the other hand, by the definition of ζ and ρ_i,

$$\begin{aligned}\left|\left(\frac{1}{\rho_1}-\rho_1\right)+\zeta\left(\frac{1}{\rho_2}-\rho_2\right)\right| &= \frac{1}{\rho_1}\left|\left(1-\rho_1^2\right)+\frac{\mu_2}{\mu_1}\left(1-\rho_2^2\right)\right| \\ &= \frac{1}{\rho_1}\left|\beta_1 h_1^{\alpha}+\frac{\mu_2}{\mu_1}\beta_2 h_2^{\alpha}\right|. \end{aligned} \tag{3.7.19}$$

Thus by (3.1.6), (3.1.8) and (3.4.35), we have

$$\frac{\lambda_1}{\lambda_2}\left|\left(\frac{1}{\rho_1}-\rho_1\right)+\zeta\left(\frac{1}{\rho_2}-\rho_2\right)\right| \le C\cdot h_2^{\alpha}. \tag{3.7.20}$$

Hence the lemma follows by putting this and (3.7.18) into (3.7.17). □

Combining lemmas 3.7.1 and 3.7.2, we have, for $h_i < H_4$, $i = 1,2$,

$$\begin{aligned}\left|d_j-\left(1+\frac{\lambda_1}{\lambda_2}\right)\right| &\le \frac{C_2 h_2}{(C_1\log n_2)^2}+\frac{C_3 h_2^{\alpha-1}}{C_1\log n_2} \\ &= C_4 h_2^{\alpha_0}/\log n_2, \qquad \text{for} \quad C_1\log n_2 \le j < n_2. \end{aligned} \tag{3.7.21}$$

Here

$$\alpha_0 = \min\{\alpha-1,\,1\}, \tag{3.7.22}$$

and

$$C_4 = \frac{1}{C_1}\left(\frac{C_2}{C_1\log n_2}+C_3\right). \tag{3.7.23}$$

Hence we have

Corollary 3.7.3

For $\alpha \ge 1$ and $h_i < H_4$, $i = 1,2$,

$$D_p = \left(1+\frac{\lambda_1}{\lambda_2}\right)\cdot I + \tilde{L}_{C_1\log n_2} + U_p, \tag{3.7.24}$$

where U_p is a diagonal matrix with

$$\|U_p\|_2 < C_4 h_2^{\alpha_0}/\log n_2. \quad \square \tag{3.7.25}$$

Next we claim that the eigenvalues of W_l, $i = 1,2$, and hence of W_p, are clustered around zero. From lemma 3.6.2, the W_l are positive semi-definite matrices. Thus

$$tr\,(W_l) = \sum_{j=1}^{n_2-1} (W_l)_{jj} = \sum_{j=1}^{n_2-1} \lambda_j (W_l) \geq 0, \qquad i = 1, 2, \tag{3.7.26}$$

where $\lambda_j (W_l) \geq 0$ are the eigenvalues of W_l.

LEMMA 3.7.4

For $\alpha \geq 1$, there exists a $C_5 > 0$ and $0 < H_5 \leq H_4$, such that for all h_1 and $h_2 < H_5$,

$$tr\,(W_l) < C_5 \log n_2, \qquad i = 1, 2. \tag{3.7.27}$$

Proof By (3.6.23) and (3.6.9)

$$(W_1)_{jj} = (W_2)_{jj} = \frac{1}{\rho_2}\,\frac{2}{n_2} \sin^2 \psi_{2j} \sum_{l=1}^{n_1-1} \frac{\tilde{\gamma}_l\,(^1q_{n_1 l})^2}{(\tilde{\gamma}_l + \gamma_j)^2}, \quad 1 \leq j < n_2. \tag{3.7.28}$$

Using (2.3.21) - (2.3.24) and (3.6.17), it is easy to verify that

$$(^1q_{n_1 l})^2 = \frac{1}{\rho_1^2}\,\frac{2}{n_1} \sin^2 \psi_{1l} = \frac{2}{n_1}\,\frac{\sin^2 \theta_{1l}}{\rho_1\,\zeta\,\tilde{\gamma}_l}, \qquad 1 \leq l < n_1.$$

Thus (3.7.28) becomes

$$(W_l)_{jj} = \frac{4}{\zeta\,\rho_1\,\rho_2\,n_1\,n_2} \sin^2 \psi_{2j} \sum_{l=1}^{n_1-1} \frac{\sin^2 \theta_{1l}}{(\tilde{\gamma}_l + \gamma_j)^2}, \qquad 1 \leq j < n_2. \tag{3.7.29}$$

By (3.4.41), $\gamma_j \geq \frac{4j^2}{n_2^2}$. Similarly, by (3.6.17), $\tilde{\gamma}_l \geq \frac{4}{\zeta}\,\frac{l^2}{n_1^2}$. Using these and the fact that $\sin \theta_{1l} = \sin \frac{l\pi}{n_1} \leq \frac{l\pi}{n_1}$ and $\sin^2 \psi_{2j} \leq 1$, (3.7.29) becomes

$$(W_l)_{jj} \leq \frac{\zeta\,\pi^2}{4\,\rho_1\,\rho_2\,n_1\,n_2} \sum_{l=1}^{n_1-1} \frac{(\frac{l}{n_1})^2}{[\,\zeta\,(\frac{j}{n_2})^2 + (\frac{l}{n_1})^2\,]^2}, \qquad 1 \leq j < n_2. \tag{3.7.30}$$

Let $y_j^2 = \zeta\,(\frac{j}{n_2})^2$. Consider the function $f_j(x) = \frac{x^2}{(y_j^2 + x^2)^2}$. The maximum of $f_j(x)$ for $x \geq 0$ is at $x = y_j$ where $f_j(y_j) = (2\,y_j)^{-2}$. Hence for $1 \leq j < n_2$,

$$(W_l)_{jj} \le \frac{\zeta \pi^2}{4 \rho_1 \rho_2 n_2} \left\{ \int_0^1 \frac{x^2}{(y_j^2 + x^2)^2} dx + \frac{2}{n_1} \max_{(0,1)} \frac{x^2}{(y_j^2 + x^2)^2} \right\}$$

$$\le \frac{\zeta \pi^2}{4 \rho_1 \rho_2 n_2} \left\{ \frac{\pi}{4 y_j} + \frac{1}{2 n_1 y_j^2} \right\}$$

$$= \frac{\zeta \pi^2}{4 \rho_1 \rho_2 n_2} \left\{ \frac{n_2 \pi}{4 j \sqrt{\zeta}} + \frac{n_2^2}{2 \zeta n_1 j^2} \right\} = \frac{\sqrt{\zeta} \pi^3}{16 \rho_1 \rho_2 j} + \frac{C_0 \pi^2}{8 \rho_1 \rho_2 j^2},$$

where C_0 is given in (3.1.6). Hence for $i = 1,2$,

$$tr(W_l) \le \frac{\sqrt{\zeta} \pi^3}{16 \rho_1 \rho_2} \sum_{j=1}^{n_2-1} j^{-1} + \frac{C_0 \pi^2}{8 \rho_1 \rho_2} \sum_{j=1}^{n_2-1} j^{-2},$$

$$\le \frac{\sqrt{\zeta} \pi^3}{16 \rho_1 \rho_2} (\log n_2 + \gamma) + \frac{C_0 \pi^2}{8 \rho_1 \rho_2} \sum_{j=1}^{n_2-1} j^{-2},$$

where γ is the Euler constant. Using the Euler formula

$$\sum_{j=1}^{n_2-1} j^{-2} < \sum_{j=1}^{\infty} j^{-2} = \frac{\pi^2}{6},$$

we have

$$tr(W_l) \le \frac{\pi^3}{16 \rho_1 \rho_2} \left\{ \sqrt{\zeta} \log n_2 + \gamma + \frac{\pi C_0}{3} \right\}, \qquad i=1,2. \tag{3.7.31}$$

Thus there exists a C_5 such that (3.7.27) holds for all sufficiently large n_2. □

From this lemma, we see that the number of eigenvalues of W_l that are larger than 1 cannot exceed $C_5 \log n_2$. Since the W_l are symmetric, they can be diagonalized. By separating those eigenvalues that are greater than 1 from those that are smaller, we have

Collorary 3.7.5

If $\alpha \ge 1$ and $h_1, h_2 < H_5$, then

$$W_l = V_l + \tilde{L}^l_{C_5 \log n_2}, \qquad i = 1,2. \tag{3.7.32}$$

Here the V_l are positive semi-definite matrices with $\| V_l \|_2 < 1$, or equivalently,

$$0 \le \frac{x^* V_l x}{x^* x} \le 1, \qquad \text{for } x \in R^{n_2 - 1}, \tag{3.7.33}$$

and the $\tilde{L}^l_{C_5 \log n_2}$ are positive definite matrices with

$$\lambda_{\min} (\tilde{L}^l_{C_5 \log n_2}) \ge 1. \quad \square \tag{3.7.34}$$

Define the constant

$$C_6 = 2 C_5 + C_1. \tag{3.7.35}$$

By (3.7.24) and (3.7.32), we get

$$D_p + \frac{\lambda_1}{\lambda_2} (W_2 - W_1) = I + \frac{\lambda_1}{\lambda_2} (I + V_2 - V_1) + \tilde{L}_{C_6 \log n_2} + U_p, \tag{3.7.36}$$

where

$$\tilde{L}_{C_6 \log n_2} = \tilde{L}^2_{C_5 \log n_2} - \tilde{L}^1_{C_5 \log n_2} + \tilde{L}_{C_1 \log n_2}. \tag{3.7.37}$$

Using (3.7.33), we see that, the eigenvalues of $I + \frac{\lambda_1}{\lambda_2} (I + V_2 - V_1)$ lie in the interval $[1 , 1 + 2 \frac{\lambda_1}{\lambda_2}]$. Thus by (3.7.7) and (3.7.25), we have

Corollary 3.7.6

If $\alpha \ge 1$ and $h_l < H_5$, $i = 1,2$, then

$$B_s^* B_s = V_s + U_s + \tilde{L}_{C_6 \log n_2 + 8}. \tag{3.7.38}$$

Here

$$V_s = \begin{bmatrix} I + \frac{\lambda_1}{\lambda_2} (I + V_2 - V_1) & \mathbf{0}_{n_2 - 1} \\ \mathbf{0}^*_{n_2 - 1} & 1 + \frac{\lambda_1}{\lambda_2} \end{bmatrix} \tag{3.7.39}$$

is symmetric with

$$\lambda_j (V_s) \in [1 , 1 + 2 \frac{\lambda_1}{\lambda_2}], \qquad 1 \le j \le n_2, \tag{3.7.40}$$

$$U_s = \begin{bmatrix} U_p & \mathbf{0}_{n_2-1} \\ \mathbf{0}^*_{n_2-1} & 0 \end{bmatrix} \tag{3.7.41}$$

is diagonal with

$$\| U_s \|_2 < C_4 \, h_2^{\alpha_0} / \log n_2, \tag{3.7.42}$$

and

$$\tilde{L}_{C_6 \log n_2 + 8} = \tilde{L}_{C_6 \log n_2} + \tilde{L}_8. \quad \square \tag{3.7.43}$$

From (3.1.8), (3.7.40) and (3.7.42), we see that there exists d_1 and $d_2 > 0$, independent of h_l, such that for all h_l sufficiently small, all the eigenvalues of $V_s + U_s$ lie in the interval $[\, d_1 , d_2 \,]$. Notice that the matrices V_s and U_s are symmetric, hence $\tilde{L}_{C_6 \log n_2 + 8}$ is also symmetric. Clearly, $\tilde{L}_{C_6 \log n_2 + 8}$ can be written as a sum of $(C_6 \log n_2 + 8)$'s symmetric rank one matrices. Thus using the Cauchy interlace theorem (see Parlett [34]) repeatedly for $(C_6 \log n_2 + 8)$ times, we get

Corollary 3.7.7

For $\alpha \geq 1$, there exists d_1, $d_2 > 0$, and $H_6 < H_5$ such that for h_1, $h_2 < H_6$, all except $(C_6 \log n_2 + 8)$ eigenvalues of $B_s^* B_s$ lie in the interval $[\, d_1 , d_2 \,]$. $\square$

Using lemma 3.3.2, we immediate get

Corollary 3.7.8

For $\alpha \geq 1$, there exists b_1 and $b_2 > 0$, independent of h_l, such that for h_1, $h_2 < H_6$, all except $(C_6 \log n_2 + 8)$ eigenvalues of $B^* B$ lie in the interval $[\, b_1 , b_2 \,]$. $\square$

Using this corollary, we are able to derive an upper bound for the number of iterations required to attain a given accuracy.

§ 3.8 The Rate of Convergence when $\alpha \geq 1$

In this section, we will show that the number of iterations required to attain a given accuracy increases no faster than $O(\log^2 n_2)$. We first derive an upper bound for (2.2.3).

LEMMA 3.8.1

Let x be the solution to $B^* B\, x = B^* b$ and x_j be the j-th iterant of the ordinary conjugate gradient method (2.2.1) applied to this normal equation. If the eigenvalues $\{ \delta_j \}$ of $B^* B$ are such that

$$0 < \delta_1 \leq \cdots \leq \delta_{p-1} < \delta_p = b_1 \leq \cdots \leq \delta_{n_2-q} = b_2 < \delta_{n_2-q+1} \leq \cdots \leq \delta_{n_2},$$

then

$$\frac{\| B(x-x_j) \|_2}{\| B(x-x_0) \|_2} \leq 2 \left(\frac{b-1}{b+1} \right)^{j-p-q} \cdot \max_{\delta \in [b_1, b_2]} \left\{ \prod_{j=1}^{p} \left(\frac{\delta - \delta_j}{\delta_j} \right) \prod_{j=n_2-q+1}^{n_2} \left(\frac{\delta_j - \delta}{\delta_j} \right) \right\}. \quad (3.8.1)$$

Here

$$b = \left(\frac{b_2}{b_1} \right)^{\frac{1}{q}} \geq 1. \quad \square \quad (3.8.2)$$

The proof can be found in Van der Vorst [39], see also Daniel [13].

Notice that for $n_2 - q + 1 \leq j \leq n_2$ and $\delta \in [\, b_1 , b_2 \,]$, we have,

$$0 \leq \frac{\delta_j - \delta}{\delta_j} \leq 1 - \frac{\delta}{\delta_j} \leq 1. \quad (3.8.3)$$

Thus (3.8.1) can be rewritten as

$$\frac{\| B(x-x_j) \|_2}{\| B(x-x_0) \|_2} \leq 2 \left(\frac{b-1}{b+1} \right)^{j-p-q} \cdot \max_{\delta \in [b_1, b_2]} \prod_{j=1}^{p} \left(\frac{\delta - \delta_j}{\delta_j} \right). \quad (3.8.4)$$

From lemmas 3.5.5 and 3.3.2, we see that if $\alpha > 1$ or if $\alpha = 1$ such that (3.5.10) holds, then

$$\delta_j \geq C \cdot h_2^6, \qquad 1 \leq j \leq n_2. \quad (3.8.5)$$

Thus for $1 \leq j \leq p$ and $\delta \in [\, b_1 , b_2 \,]$, we have,

$$0 \le \frac{\delta - \delta_j}{\delta_j} \le C\, n_2^{\delta}. \tag{3.8.6}$$

Hence (3.8.4) becomes

$$\frac{\| B(x-x_j) \|_2}{\| B(x-x_0) \|_2} \le C^p\, n_2^{\delta p} \cdot \left(1 - \frac{2}{b+1}\right)^{j-p-q} = C^p\, n_2^{\delta p} \cdot e^{(j-p-q)\log\left(1-\frac{2}{b+1}\right)}.$$

Since $\log(1-x) \le -x$ for $0 \le x \le 1$, we have

$$\frac{\| B(x-x_j) \|_2}{\| B(x-x_0) \|_2} \le C^p\, n_2^{\delta p} \cdot e^{(p+q-j)\frac{2}{b+1}} = e^{p\log C + \delta p \log n_2 + (p+q-j)\frac{2}{b+1}}. \tag{3.8.7}$$

Thus given arbitrary $\epsilon > 0$, an upper bound for the number of iterations required to make

$$\frac{\| B(x-x_j) \|_2}{\| B(x-x_0) \|_2} \le \epsilon$$

is given by

$$j_0 = \frac{b+1}{2}\{\, p\log C + 6\, p\log n_2 - \log \epsilon \,\} + p + q. \tag{3.8.8}$$

Notice that by corollary 3.7.8, $b = \dfrac{b_2}{b_1}$ is independent of n_2. Moreover, we have

$$p + q \le C_6 \log n_2 + 8 \le (2\, C_5 + C_1) \log n_2 + 8. \tag{3.8.9}$$

Using (3.7.34) and (3.7.37), we also have

$$p\ ,\ q \le (C_5 + C_1) \log n_2 + 8. \tag{3.8.10}$$

Thus the right hand side of (3.8.8) is bounded above by $C \cdot \log^2 n_2$. Hence we have,

THEOREM 3.8.2

Assume that either $\alpha = 1$ and (3.5.10) holds or $\alpha > 1$, then the number of iterations required to reduce $\dfrac{\| B(x-x_j) \|_2}{\| B(x-x_0) \|_2}$ by a given accuracy can grow no faster than $O(\log^2 n_2)$ as h_l tends to zero. □

§ 3.9 Concluding Remarks

In this section, we will generalize the results in the previous sections to the case where $\lambda_l \geq \mu_l$. We will also discuss the multi-server case where $s_l > 1$. Let us begin with the case where $\lambda_l > \mu_l$.

§ 3.9.1 The Case when $\lambda_l > \mu_l$ and $\alpha \geq 1$

Going over the proof in § 3.3 - § 3.8, we see that when $\lambda_l > \mu_l$, some of the arguments are no longer valid or need further explanations. They are (3.3.2) - (3.3.6), (3.4.24), (3.4.25), (3.4.33), (3.4.38), (3.5.9) and (3.7.17). We will see in the following that these equations can easily be modified to prove the same assertions. The idea is to expand ρ_l around $h_l = 0$. Let us assume

$$\frac{\lambda_l}{\mu_l} = 1 + \beta_l h_l^{\alpha} > 1, \tag{3.9.1}$$

where

$$\alpha \geq 1 \text{ and } 0 < \beta_l \leq 1. \tag{3.9.2}$$

Here α, μ_l and β_l are assumed to be constant independent of h_l. By definition (3.1.12)

$$\rho_l = \left(\frac{\lambda_l}{\mu_l} \right)^{1/2} = \left(1 + \beta_l h_l^{\alpha} \right)^{1/2} \tag{3.9.3}$$

and hence

$$\rho_l^{n_l - 1} = e^{\frac{1}{2h_l} \log (1 - \beta_l h_l^{\alpha})} \leq e^{\frac{1}{2} \beta_l h_l^{\alpha - 1}} \leq \left(1 - \tfrac{1}{2} \beta_l h_l^{\alpha - 1} \right)^{-1}. \tag{3.9.4}$$

In the following paragraphs, we will replace or give another proof for the equations listed above.

First we observe that by (3.1.11) and the fact that $\rho_2 > 1$, (3.3.2), (3.3.3) and (3.3.5) can be replaced respectively by

$$\| a_2^{-1} S_2 Q_2 \|_2 = \rho_2^{n_2 - 1}, \tag{3.9.5}$$

$$\| a_2 Q_2^* S_2^{-1} \|_2 = 1, \tag{3.9.6}$$

and

$$\rho_2^{1-n_2} \sigma_j \le \hat{\sigma}_j \le \rho_2^{n_2-1} \sigma_j, \qquad 1 \le j \le n_2. \tag{3.9.7}$$

By (3.9.4) and (3.9.7), (3.3.6) can be replaced by

$$(1 - \tfrac{1}{8} \beta_i h_i^{\alpha-1}) \sigma_j \le \hat{\sigma}_j \le (1 - \tfrac{1}{8} \beta_i h_i^{\alpha-1})^{-1} \sigma_j, \qquad 1 \le j \le n_2. \tag{3.9.8}$$

In particular, (3.3.7) still holds.

Next we claim that $\frac{\rho_1 - a_j}{\rho_1 \gamma_j}$ in (3.4.24) is a decreasing function of j for $1 \le j < n_2$.

Since $\rho_1 > 1 > a_j$, the left hand side in (3.4.24) is positive. Hence

$$1 > \rho_1 a_j, \qquad 1 \le j < n_2. \tag{3.9.9}$$

Since a_j is a decreasing function of j, we see that the right hand side in (3.4.24) is a decreasing function of j.

Similarly, $\frac{1 - a_j}{\gamma_j}$ in (3.4.25) is also a decreasing function of j. In fact, by (3.4.24)

$$\frac{1 - a_j}{\gamma_j} = \frac{\rho_1 - 1}{\rho_1 \gamma_j} + \frac{1 - \rho_1 a_j}{\rho_1 \gamma_j} = \frac{\rho_1 - 1}{\rho_1 \gamma_j} + \frac{\zeta a_j}{\rho_1 - a_j}. \tag{3.9.10}$$

Since $\rho_1 > 1 > a_j$ and a_j, γ_j^{-1} are decreasing functions of j, the assertion follows.

The first inequality in (3.4.33) follows from (3.9.9) and (3.4.32).

Instead of getting an upper bound for ρ_i^{-1} in (3.4.33), we now need an upper bound for ρ_i. By (3.9.3) and (3.9.2),

$$\rho_i \le (1 + \beta_i h_i^{\alpha}) \le 2, \qquad i = 1,2. \tag{3.9.11}$$

Next we modify (3.4.38). Notice that by (3.9.3), $\rho_1 < 1 + O(h_i^{\alpha})$. Thus

$$\frac{1}{1 - \rho_1 a_j} \le \frac{1}{1 - a_j} + O(h_i^{\alpha}), \qquad 1 \le j < n_2. \tag{3.9.12}$$

Hence (3.4.38) can be replaced by

$$\phi_j < \frac{\zeta a_j}{1 - a_j} + \frac{1}{\gamma_j n_1} + O(h_i^{\alpha}), \qquad 1 \le j < n_2. \tag{3.9.13}$$

It is obvious that the extra term $O(h_i^{\alpha})$ does not affect the conclusion of lemma 3.4.3.

Since $\rho_2 > 1 > \cos\theta_1$, thus (3.5.9) is positive for all $\alpha \geq 1$. Hence in theorem 3.5.6 and 3.8.2, there is no need to assume that (3.5.10) holds even when $\alpha = 1$.

Finally replacing (3.7.16) by (3.9.12), we see that (3.7.17) can be replaced by

$$| \tilde{d}_j - (1 + \frac{\lambda_1}{\lambda_2}) | \leq \frac{\lambda_1}{\lambda_2} \{ \frac{a_j}{1 - a_j} + O(h_l^\alpha) \} | \frac{1}{\rho_1} - \rho_1 + \zeta(\frac{1}{\rho_2} - \rho_2) |. \quad (3.9.14)$$

It is also clear that the extra term $O(h_l^\alpha)$ does not affect the conclusion of lemma 3.7.2.

Using these modified equations, it is easy to prove

THEOREM 3.9.1

If (3.9.1) and (3.9.2) hold, then the number of iterations required to reduce $\frac{\| B(x-x_j) \|_2}{\| B(x-x_0) \|_2}$ by a given accuracy can grow no faster than $O(\log^2 n_2)$ as h_l tends to zero. □

§ 3.9.2 The Case when $\lambda_l = \mu_l$

When $\lambda_l = \mu_l$, $i = 1,2$, by (3.1.12) and (3.1.3), we have

$$\rho_l = (\frac{\lambda_l}{\mu_l})^{1/4} = 1, \qquad i=1,2, \quad (3.9.15)$$

$$\beta_l = 0, \qquad i=1,2, \quad (3.9.16)$$

and α is arbitrary. Moreover, by the definition (2.1.7) and (2.1.8),

$$a_l = (\frac{1}{n_l})^{1/4}, \qquad i=1,2. \quad (3.9.17)$$

Using (3.9.15) - (3.9.17), it is straightforward to check that all the results in § 3.3 - § 3.8 are still valid. In particular, theorem 3.8.2 still holds in this case.

We remark that when $\lambda_l = \mu_l$, $i = 1,2$, we are in fact preconditioning an oblique BVP by a Neumann BVP. To see this, we first observe that by (2.1.5) and the fact that $\lambda_l = \mu_l$,

$$A_0 = \lambda_1 (\bar{G}_1 \otimes I_{n_2}) + \lambda_2 (I_{n_1} \otimes \bar{G}_2), \quad (3.9.18)$$

where the $\bar{G}_l$ are of order n_l and are given by

$$\bar{G}_l = \text{tridiag} (-1 , 2 , -1) - e_1 e_1^* - e_{n_l} e_{n_l}^*, \qquad i=1,2. \quad (3.9.19)$$

Thus A_0 is the 5-point formula for the Neumann problem

$$\begin{cases} \lambda_1 \dfrac{\partial^2 p}{\partial x^2} + \lambda_2 \dfrac{\partial^2 p}{\partial y^2} = 0 & \text{in } [0,1]^2, \\ \dfrac{\partial p}{\partial \eta} = 0 & \text{on } \partial[0,1]^2, \end{cases} \tag{3.9.20}$$

with mesh-size $h_i = (n_i - 1)^{-1}$, $i = 1,2$. Here η denotes the unit outward normal of the square $[0,1]^2$. On the other hand, from (2.3.6) and (3.4.1), we see that

$$R_0 = ({}^1e_{n_1}\,{}^1e_{n_1}^*) \otimes {}^2R_1 = ({}^1e_{n_1}\,{}^1e_{n_1}^*) \otimes (\lambda_1 \cdot R_x), \tag{3.9.21}$$

where R_x is given by (3.4.1) with $\rho_2 = 1$ there. Thus R_x is a forward difference approximation of ∂_x. It is easy to see that

$$A = A_0 + R_0 \tag{3.9.22}$$

is a 5-point formula for the oblique BVP

$$\begin{cases} \lambda_1 \dfrac{\partial^2 p}{\partial x^2} + \lambda_2 \dfrac{\partial^2 p}{\partial y^2} = 0 & \text{in } [0,1]^2, \\ \dfrac{\partial p}{\partial \gamma} = 0 & \text{on } \partial[0,1]^2. \end{cases} \tag{3.9.23}$$

Here γ is a directional vector defined on $\partial[0,1]^2$ and is given by

$$\gamma = \begin{cases} \eta & \text{if } x \neq 1, \\ \eta + \tau & \text{if } x = 1, \end{cases} \tag{3.9.24}$$

where τ is the unit tangential vector.

Thus the preconditioning of A by A_0 is the discrete version of preconditioning the oblique problem (3.9.23) by the Neumann problem (3.9.20). Since theorem 3.8.2 holds in this case, the matrix A_0 is a very good preconditioner for A. Moreover, by the results in § 3.7, we see that the singular values of the preconditioned matrix

$$A\,A_0^+ = I + R_0\,A_0^+ \tag{3.9.25}$$

are clustered around $(1 + \dfrac{\lambda_1}{\lambda_2})^{\frac{1}{4}}$.

§ 3.9.3 The Multi-Server Case

For $s_l > 1$, then instead of (2.3.28), the underlying continous equation is of the form

$$(\lambda_1 + s_1 \mu_1) p_{xx} + (\lambda_2 + s_2 \mu_2) p_{yy} + 2 h_1 (s_1 \mu_1 - \lambda_1) p_x + 2 h_2 (s_2 \mu_2 - \lambda_2) p_y \approx 0, \tag{3.9.26}$$

in the region where $s_1 h_1 \le x = i h_1 \le 1$ and $s_2 h_2 \le y = j h_2 \le 1$. In other part of the square $[0,1]^2$, the equation has variable coefficients. Thus if s_l are constant independent of h_l, then for sufficiently small h_l, a sensible limit to consider is

$$s_l \mu_l = \lambda_l \pm \eta_l h_l^{\alpha}, \qquad i = 1,2, \tag{3.9.27}$$

for some constants η_l and α. We remark that when $s_l = 1$ and $\eta_l \le \mu_l$, $i = 1,2$, this reduces to the limit we discussed previously.

When s_l increases proportionally to n_l, (3.9.27) may not be the right limit to consider. We remark that in this case, the diagonal matrix S_l that is used in the transformation (3.3.1) is no longer well-conditioned. In fact, by (2.1.7), the last diagonal entry of S_l is given by

$${}^{l}d_{n_l} = a_l \cdot \left[\frac{1}{s_l !} \left(\frac{\lambda_l}{\mu_l}\right)^{s_l} \left(\frac{\lambda_l}{s_l \mu_l}\right)^{n_l - s_l - 1} \right]^{\frac{1}{4}}, \qquad i = 1,2. \tag{3.9.28}$$

Thus by (3.9.27) and Stirling's formula, we have, for $\alpha \ge 1$,

$${}^{l}d_{n_l} \approx a_l \cdot \left[\frac{(s_l)^{s_l}}{s_l !} \right]^{\frac{1}{4}} \approx C\, s_l^{\frac{1}{4}} e^{-\frac{1}{2} s_l}, \qquad i = 1,2. \tag{3.9.29}$$

Thus the condition number of S_l increases exponentially for all $\alpha \ge 1$.

We remark that the ordinary conjugate gradient method (2.2.1) indeed converges within a few steps when s_l = constant, while it diverges when $s_l = n_l - 1$, see § 4.

This concludes our discussion on the model problems.

Section 4. Numerical Results

In this section, we report on the numerical results for the models discussed in § 2. All the computations were carried out on the Cyber-760 at the Mathematics and Computing Laboratory of the Courant Institute. Single precision, between fourteen and fifteen decimal digits, was used throughout. Craig's method used in these computations is a version of the ordinary conjugate gradient method (2.2.1) applied to the normal equations; see Elman [15]. For a given tolerance, convergence is said to occur at the k-th step if

$$\frac{\| r_k \|_2}{\| r_0 \|_2} \leq \text{tolerance},$$

while

$$\frac{\| r_{k-1} \|_2}{\| r_0 \|_2} > \text{tolerance}.$$

Here r_k, given by (2.2.1) or (2.2.5), is the residual at the k-th step and

$$\| x \|_2^2 = \frac{1}{m} \sum_{i=1}^{m} x_i^2 \qquad \text{for all } x \in R^m.$$

The initial iterant x_0 is chosen to be identically zero.

(I) Two-queue, one-direction overflow model.

This is the model discussed in § 2.3.1. Iterative methods developed in § 2.2 are used to solve the matrix equation $B y_0 = b$. Here B and b are given by (2.3.12) and (2.3.13) respectively. Let us first consider the case where $\lambda_i < \mu_i$ and $s_i = 1$. Tables 1a, 1b, 2a and 2b give the number of iterations required to converge for two different choices of β_i, where β_i is the parameter defined by (3.1.3) or (3.9.1). We see that the convergence rate is

independent of β_l. The results also show that the case where $\alpha = 1$ is critical for Craig's method. More precisely, when $\alpha > 1$, the number of iterations seems to be constant as $n_l \to \infty$. When $\alpha = 1$, it increases like $O(\log n_l)$. This is consistent with (3.7.22). When $\alpha < 1$, Craig's method does not converge for sufficiently large n_l. However, from theorem 3.2.1, we see that for sufficiently large n_l, there is no need to solve the matrix equation numerically.

Next we consider the case where $\lambda_l \geq \mu_l$ and $s_l = 1$. Tables 3a and 3b give the results of our method for different choices of α. In the tables, $\alpha = \infty$ represents the problems where $\lambda_l = \mu_l$. We see that the convergence rate in this case is almost the same as in the case where $\lambda_l < \mu_l$. We note that in all the cases considered, our method converges much sooner than the bound given in (3.8.8). Tables 4 and 5 provide an explanation. In tables 4a and 4b, $\{\delta_j\}_{j=1}^{32}$ are the eigenvalues of the iteration matrix $B\,B^*$ arranged in ascending order. We see that the δ_j are more clustered than suggested by our results in § 3.7. In tables 5a and 5b we give the largest and smallest eigenvalues of $B\,B^*$ for increasing values of n_l. We note that $\kappa(B)$ goes to infinity because of the presence of some small and some large singular values. However, $\kappa(B)$ is much smaller than the bound obtained in theorem 3.5.6.

Tables 6a and 6b give the time required for the different stages of the algorithms. "Initialization" refers to the generation of Φ, S_l, Q_l and the right hand side b. "Iteration" refers to the solving of $By_0 = b$ by the iterative methods. "Generating p" refers to the computation of p in (2.3.14). Alternative C in Appendix A.1 is used in computing the matrix-vector product Bd in each iteration. We note that the timings are consistent with the theoretical estimates. In fact, from § 2.3.1, we see that the work and storage requirement per iteration are $O(n_l \log n_l)$ and $O(n_l)$ repectively. The generation of p requires an extra $O(n_l^2 \log n_l)$ work and n_l^2 storage. We remark that substantial saving would result if only a few entries of p are needed.

Tables 7a and 7b give the results for a family of multi-server problems that satisfy (3.9.27). As remarked there, when $s_l = O(n_l)$, (3.9.27) may not be a good limit. Indeed,

Craig's method diverges when $s_l = n_l - 1$. We note that in the multi-server case, $s_l > 1$, we cannot use the Fast Fourier Transform. Hence the work and storage requirement per iteration are $2n_l^2 + O(n_l)$ and $n_l^2 + O(n_l)$ respectively.

Let us compare our method with other conventional methods. As mentioned in § 2.4, a typical approach to this kind of queueing problems is to fix one degree of freedom in the solution p. Or equivalently, one row in the generating matrix A is deleted and the resulting nonsingular system solved. Let us consider the case where this nonsingular system is solved by a classical iterative method such as the point SOR method, see Kaufman [25]. Since the graph of the generating matrix A is the same as the graph of a discrete Laplacian, it is clear that the point SOR method requires $7\,n_l^2 + O(n_l)$ work and $n_l^2 + O(n_l)$ storage spaces per iteration. We note that this method converges very slowly. Tables 8 and 9 give the numerical evidences. Table 8 lists the number of iterations required by the two methods and table 9 compares the time required in seconds. In the tables, ω^* denotes the optimal relaxation factor up to three decimal points obtained experimentally. We see that the point SOR method has a very slow convergence rate especially when s_l is small. In the three cases we considered, our method converges 5 to 32 times faster than the point SOR method.

Let us now consider the approach of solving the nonsingular system by a direct method. Since the band-width of the generating matrix A is n_l, the band Gaussian elimination will require $O(n_l^4)$ work and $O(n_l^3)$ storage spaces. A direct method that takes advantage of the separablity of the problem will reduce these counts to $O(n_l^3)$ and $O(n_l^2)$ respectively, see Kaufman [25]. Since the graph of A is the same as the graph of a discrete Laplacian, nested dissection method can also be used, see George and Liu [18]. The counts for this method are $O(n_l^3)$ and $O(n_l^2 \log n_l)$ respectively. Let us remark that our preconditioned system $B\,y_0 = b$ can also be solved by direct methods. In fact, we can compute and store B by using (2.3.17). This would require $O(n_l^3)$ operations and $O(n_l^2)$ storage spaces. Thus we see that solving the preconditioned system $B\,y_0 = b$ by conjugate gradient type methods requires the least amount of work and storage.

(II) Other Separable Preconditioners

Tables 10a and 10b report on the performance of a family of preconditioners. The parameter β in the tables indicates the preconditioners we are using. If $\beta \neq 0$, the preconditioner is $\tilde{A}_\beta$ which is nonsingular and is given by (2.4.5). If $\beta = 0$, the preconditioner is defined by (2.4.6). When $\beta = A_0$ the preconditioner is A_0 itself. Recall that when $\beta = 1$, the preconditioner resembles a Dirichlet problem while when $\beta = 0$, it resembles a Neumann problem. We see that the number of iterations decreases as $|\beta| \to 0$. However, we remark that for sufficiently small β, arithmetic overflow will occur. This is because by (2.4.4) and (2.4.5), the smallest eigenvalue of $\tilde{A}_\beta$ tends to zero as $|\beta| \to 0$.

(III) Numerical Results for other queueing models

Tables 11a and 11b give the number of iterations required for convergence for the model we discussed in § 2.3.2, i.e., the 2-queue model with overflow permitted in both directions. We see that our method does not converge when $s_l = n_l - 1$. However, when the s_l are constant and independent of n_l, the performance is quite good. The number of iterations seems to increase proportionally to $\log n_l$ and the convergence rate is almost independent of α.

Tables 12 gives the results of our method when applied to the 3-queue model discussed in § 2.5.1. The number of iterations also increases like order $O(\log n_l)$. Table 13 shows the time required in each phase of the algorithm. The time per iteration is slightly better than the theoretical estimates obtained in Appendix A.3, see (A.3.16). Tables 14 and 15 compare our method with the point SOR method. We see that our method performs much better than the point SOR method especially when the s_l are small.

Finally, we consider the model discussed in § 2.6. Tables 16a and 16b give the number of iterations required for convergence using the method introduced in § 2.6.1. In all cases, we see that the number of iterations increases at most like $O(\log n_l)$.

For easy reference, in the following we use the notation Ma.b.c to denote the method we developed in § a.b.c. For the definitions of β_l, η_l and α, see (3.1.3) and (3.9.27).

Table 1a. M2.3.1 (tolerance = 10^{-10})

Craig's Method: $\frac{\lambda_l}{\mu_l} = 1 - \beta_l h_l^{\alpha}$, $\mu_l = s_l = 1$, $\beta_l = \frac{1}{8}$, $i=1,2$

(n_1,n_2)	α				(n_1,n_2)	α				(n_1,n_2)	α			
	0.	1.	2.	3.		0.	1.	2.	3.		0.	1.	2.	3.
(5,8)	8	8	7	7	(8,8)	8	8	7	7	(8,5)	5	5	5	5
(10,16)	13	11	9	9	(16,16)	13	10	8	8	(16,10)	10	8	8	7
(20,32)	15	11	10	10	(32,32)	15	11	10	10	(32,20)	13	11	9	9
(40,64)	22	13	12	12	(64,64)	22	13	12	12	(64,40)	15	12	11	11
(80,128)	**	15	12	12	(128,128)	**	15	12	12	(128,80)	**	13	12	12

** more than 30 iterations.

Table 1b. M2.3.1 (tolerance = 10^{-10})

Orthodir Method: $\frac{\lambda_l}{\mu_l} = 1 - \beta_l h_l^{\alpha}$, $\mu_l = s_l = 1$, $\beta_l = \frac{1}{8}$, $i=1,2$

(n_1,n_2)	α				(n_1,n_2)	α				(n_1,n_2)	α			
	0.	1.	2.	3.		0.	1.	2.	3.		0.	1.	2.	3.
(5,8)	7	7	7	7	(8,8)	7	7	7	7	(8,5)	4	4	4	4
(10,16)	14	15	15	15	(16,16)	14	15	15	15	(16,10)	9	9	9	9
(20,32)	15	21	21	21	(32,32)	15	21	21	21	(32,20)	15	17	17	17
(40,64)	15	24	25	25	(64,64)	15	24	25	25	(64,40)	15	22	22	22
(80,128)	15	27	27	27	(128,128)	15	27	27	27	(128,80)	15	25	25	25

Table 2a. M2.3.1 (tolerance = 10^{-10})

Craig's Method: $\frac{\lambda_i}{\mu_i} = 1 - \beta_i h_i^{\alpha}$, $\mu_i = s_i = \beta_i = 1$, $i=1,2$											
(n_1,n_2)	α			(n_1,n_2)	α			(n_1,n_2)	α		
	1.	2.	3.		1.	2.	3.		1.	2.	3.
(5,8)	8	8	7	(8,8)	8	7	7	(8,5)	5	5	5
(10,16)	11	8	8	(16,16)	11	9	9	(16,10)	9	8	7
(20,32)	12	10	10	(32,32)	12	10	10	(32,20)	11	9	9
(40,64)	14	12	12	(64,64)	14	12	12	(64,40)	13	11	11
(80,128)	16	12	12	(128,128)	16	12	12	(128,80)	15	12	12

Table 2b. M2.3.1 (tolerance = 10^{-10})

Orthodir Method: $\frac{\lambda_i}{\mu_i} = 1 - \beta_i h_i^{\alpha}$, $\mu_i = s_i = \beta_i = 1$, $i=1,2$											
(n_1,n_2)	α			(n_1,n_2)	α			(n_1,n_2)	α		
	1.	2.	3.		1.	2.	3.		1.	2.	3.
(5,8)	7	7	7	(8,8)	7	7	7	(8,5)	4	4	4
(10,16)	15	15	15	(16,16)	15	15	15	(16,10)	9	9	9
(20,32)	20	21	21	(32,32)	21	21	21	(32,20)	17	17	17
(40,64)	24	25	25	(64,64)	24	24	24	(64,40)	22	22	22
(80,128)	26	27	27	(128,128)	27	27	27	(128,80)	26	25	25

Table 3a. M2.3.1 (tolerance = 10^{-10})

Craig's Method: $\frac{\lambda_l}{\mu_l} = 1 + \beta_l h_l^{\alpha}$, $\mu_l = s_l = 1$, $\beta_l = \frac{1}{2}$, $i=1,2$

(n_1,n_2)	α				(n_1,n_2)	α				(n_1,n_2)	α			
	1.	2.	3.	∞		1.	2.	3.	∞		1.	2.	3.	∞
(5;8)	8	7	7	6	(8,8)	8	7	7	6	(8,5)	5	5	5	4
(10,16)	11	8	9	7	(16,16)	10	8	9	7	(16,10)	8	8	7	6
(20,32)	13	10	10	9	(32,32)	11	10	10	9	(32,20)	11	9	9	8
(40,64)	13	12	12	11	(64,64)	13	12	12	11	(64,40)	12	11	11	9
(80,128)	15	12	12	11	(128,128)	15	12	12	11	(128,80)	13	12	12	11

Table 3b. M2.3.1 (tolerance = 10^{-10})

Orthodir Method: $\frac{\lambda_l}{\mu_l} = 1 + \beta_l h_l^{\alpha}$, $\mu_l = s_l = 1$, $\beta_l = \frac{1}{2}$, $i=1,2$

(n_1,n_2)	α				(n_1,n_2)	α				(n_1,n_2)	α			
	1.	2.	3.	∞		1.	2.	3.	∞		1.	2.	3.	∞
(5,8)	7	7	7	7	(8,8)	7	7	7	7	(8,5)	4	4	4	4
(10,16)	15	15	15	15	(16,16)	15	15	15	15	(16,10)	9	9	9	9
(20,32)	21	21	21	21	(32,32)	21	21	21	21	(32,20)	17	17	17	17
(40,64)	25	25	25	25	(64,64)	24	25	25	24	(64,40)	22	22	22	22
(80,128)	27	27	27	27	(128,128)	27	27	27	27	(128,80)	25	25	25	25

Table 4a. Eigenvalues δ_j of $B\,B^*$ when $\lambda_i < \mu_i$.

$\frac{\lambda_i}{\mu_i} = 1 - \beta_i h_i^{\alpha}$, $s_i = 1$, $\beta_i = \frac{1}{8}$, $n_i = 32$, $i=1,2$						
λ_1	1	1	1	1	4	4
λ_2	1	1	4	4	1	1
α	1	2	1	2	1	2
$1 + \frac{\lambda_1}{\lambda_2}$	2	2	1.25	1.25	5	5
δ_1	0.0594	0.0528	0.1984	0.1860	0.0240	0.0210
δ_2	0.9999	1.0000	0.9239	0.9733	1.0000	1.0000
δ_3	1.5160	1.6042	1.0001	1.0000	4.0748	4.4442
δ_4	1.8872	1.9798	1.2137	1.2424	4.5056	4.9689
δ_5	1.9181	1.9972	1.2256	1.2492	4.6062	4.9896
δ_6	1.9389	1.9981	1.2322	1.2495	4.7035	4.9907
δ_{10}	1.9687	1.9991	1.2420	1.2498	4.8340	4.9951
δ_{15}	1.9807	1.9994	1.2456	1.2499	4.8897	4.9968
δ_{20}	1.9858	1.9996	1.2470	1.2499	4.9144	4.9975
δ_{26}	1.9883	1.9996	1.2477	1.2499	4.9271	4.9978
δ_{27}	1.9885	1.9997	1.2477	1.2499	4.9282	4.9978
δ_{28}	1.9887	1.9997	1.2478	1.2499	4.9290	4.9978
δ_{29}	1.9888	1.9997	1.2478	1.2499	4.9296	4.9998
δ_{30}	1.9889	2.0236	1.2478	1.2567	4.9300	5.0961
δ_{31}	5.0099	5.6095	2.0014	2.1362	17.5431	20.8104
δ_{32}	15.5322	16.7475	5.1336	5.1793	59.3351	73.0058

Table 4b. Eigenvalues δ_j of $B\,B^*$ when $\lambda_i \geq \mu_i$.

$\frac{\lambda_i}{\mu_i} = 1 + \beta_i\, h_i^{\alpha}$, $s_i = 1$, $\beta_i = \frac{1}{2}$, $n_i = 32$, $i=1,2$							
λ_1	1	1	1	1	1	4	4
λ_2	1	1	1	4	4	1	1
α	∞	1	2	1	2	1	2
$1 + \frac{\lambda_1}{\lambda_2}$	2	2	2	1.25	1.25	5	5
δ_1	0.0526	0.0454	0.0523	0.1696	0.1851	0.0180	0.0208
δ_2	1.0000	0.9999	1.0000	0.9996	0.9758	1.0000	1.0000
δ_3	1.6069	1.6893	1.6096	1.0136	1.0000	4.7318	4.4646
δ_4	1.9820	2.0110	1.9841	1.2522	1.2438	5.0699	4.9856
δ_5	1.9993	2.0111	2.0003	1.2522	1.2501	5.0703	5.0022
δ_6	2.0000	2.0112	2.0003	1.2522	1.2501	5.0709	5.0022
δ_{10}	2.0000	2.0121	2.0004	1.2524	1.2501	5.0759	5.0023
δ_{15}	2.0000	2.0149	2.0004	1.2531	1.2501	5.0900	5.0027
δ_{20}	2.0000	2.0209	2.0006	1.2548	1.2501	5.1203	5.0036
δ_{26}	2.0000	2.0427	2.0013	1.2613	1.2503	5.2297	5.0070
δ_{27}	2.0000	2.0513	2.0016	1.2639	1.2504	5.2773	5.0086
δ_{28}	2.0000	2.0644	2.0020	1.2679	1.2505	5.3408	5.0115
δ_{29}	2.0009	2.0869	2.0030	1.2747	1.2508	5.4727	5.0213
δ_{30}	2.0260	2.1308	2.0285	1.2879	1.2582	5.7004	5.1254
δ_{31}	5.6294	6.2678	5.6493	2.2865	2.1452	24.5105	21.0307
δ_{32}	16.7993	18.9737	16.8519	5.3911	5.1870	95.9518	74.1012

Table 5a. Smallest Eigenvalue δ_1 of $B\,B^*$ when $\lambda_i \le \mu_i$.

$\delta_1 \times 100$		$\frac{\lambda_i}{\mu_i} = 1 - \beta_i h_i^{\alpha}$, $s_i = 1$, $\beta_i = \frac{1}{4}$, $i=1,2$						
λ_1		1	1	1	1	1	4	4
λ_2		1	1	1	4	4	1	1
α		∞	1	2	1	2	1	2
n_i	8	48.6207	48.9467	48.7007	65.7717	65.9378	44.2088	44.9619
	16	17.7985	19.1492	17.8906	39.4756	38.4127	11.3761	10.5256
	32	5.2563	5.9433	5.2783	19.8390	18.5981	2.4024	2.1027
	64	1.3788	1.5999	1.3822	8.3698	7.5914	0.4671	0.3983

Table 5b. Largest Eigenvalue δ_{n_i} of $B\,B^*$ when $\lambda_i \le \mu_i$.

δ_{n_i}		$\frac{\lambda_i}{\mu_i} = 1 - \beta_i h_i^{\alpha}$, $s_i = 1$, $\beta_i = \frac{1}{4}$, $i=1,2$						
λ_1		1	1	1	1	1	4	4
λ_2		1	1	1	4	4	1	1
α		∞	1	2	1	2	1	2
n_i	8	5.1474	4.7778	5.0950	2.0909	2.1230	15.8967	19.0733
	16	9.0321	8.3977	8.9830	3.1472	3.1792	30.3597	36.9295
	32	16.7993	15.5323	16.7475	5.1336	5.1793	59.3352	73.0058
	64	32.4994	29.7953	32.4431	8.9788	9.0954	117.5722	145.8702

Table 6a. M2.3.1 (tolerance = 10^{-10}, time in seconds)

Craig's Method: $\frac{\lambda_l}{\mu_l} = 1 - \beta_l h_l^{\alpha}$, $\mu_l = s_l = 1$, $\beta_l = \frac{1}{2}$, $i=1,2$, $\alpha = 2$					
n_l	8	16	32	64	128
Initialization	0.005	0.010	0.021	0.044	0.084
Iteration	0.097	0.201	0.473	1.066	2.283
No. of iteration	7	9	10	12	12
Time per iteration	0.0139	0.0223	0.0473	0.0888	0.1903
Generating p	0.036	0.123	0.451	1.761	7.398
Total time	0.138	0.334	0.945	2.871	9.765

Table 6b. M2.3.1 (tolerance = 10^{-10}, time in seconds)

Orthodir Method: $\frac{\lambda_l}{\mu_l} = 1 - \beta_l h_l^{\alpha}$, $\mu_l = s_l = 1$, $\beta_l = \frac{1}{2}$, $i=1,2$, $\alpha = 2$					
n_l	8	16	32	64	128
Initialization	0.007	0.012	0.023	0.045	0.093
Iteration	0.088	0.329	0.997	2.447	5.516
No. of iteration	7	15	21	25	27
Time per iteration	0.0126	0.0219	0.0475	0.0979	0.2043
Generating p	0.035	0.118	0.446	1.797	7.363
Total time	0.130	0.459	1.466	4.289	12.972

Table 7a. M2.3.1 (tolerance = 10^{-10})

Craig's Method: $s_l\,\mu_l = \lambda_l + \eta_l\, h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $i=1,2$												
s_l	1				5				n_l-1			
α	0.	1.	2.	3.	0.	1.	2.	3.	0.	1.	2.	3.
(8,8)	8	8	7	7	8	8	8	8	8	8	8	8
(16,16)	13	11	9	9	13	12	11	11	13	14	14	14
(32,32)	15	12	10	10	14	14	13	13	16	16	16	16
(64,64)	22	14	12	12	18	16	14	14	16	**	**	**
(128,128)	**	16	12	12	**	17	15	15	18	**	**	**

** more than 30 iterations

Table 7b. M2.3.1 (tolerance = 10^{-10})

Orthodir Method: $s_l\,\mu_l = \lambda_l + \eta_l\, h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $i=1,2$												
s_l	1				5				n_l-1			
α	0.	1.	2.	3.	0.	1.	2.	3.	0.	1.	2.	3.
(8,8)	7	7	7	7	7	7	7	7	7	7	7	7
(16,16)	14	15	15	15	14	15	15	15	14	13	13	13
(32,32)	15	21	21	21	16	21	21	21	16	16	16	16
(64,64)	15	24	25	24	16	25	25	25	16	18	18	18
(128,128)	15	27	27	27	16	27	27	27	16	20	20	20

Table 8. M2.3.1 (tolerance = 10^{-10})

$s_l\,\mu_l = \lambda_l + \eta_l\,h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $i=1,2$, $\alpha = 2$											
Method			point SOR: Initial guess $p_{ij} = 0$								Orthodir
n_l	s_l	N	ω^*	Relaxation factor ω							Iterations
				1.0	1.3	1.5	1.6	1.7	1.8	1.9	ω^*	
4	1	16	1.600	322	170	95	52	76	133	471	52	3
4	3	16	1.452	139	68	37	49	67	109	280	32	3
8	1	64	1.794	**	939	583	434	295	153	932	151	7
8	7	64	1.619	357	190	108	64	68	108	274	52	7

** more than 1000 iterations

Table 9. M2.3.1 (tolerance = 10^{-10}, time in seconds)

Parameter	$s_l\,\mu_l = \lambda_l + \eta_l\,h_l^{\alpha}$, $\lambda_l = s_l = 1$, $i=1,2$, $\alpha = 2$					
Problem	$n_l = 16$, $s_l = 1$		$n_l = 16$, $s_l = 15$		$n_l = 40$, $s_l = 39$	
Dimension N	256		256		1600	
Method	Orthodir	pt SOR	Orthodir	pt SOR	Orthodir	pt SOR
ω^*	...	1.891	...	1.734	...	1.833
No. of iterations	15	403	13	78	16	130
Time for iteration	0.331	14.894	0.353	2.967	2.138	30.596
Time per iteration	0.0220	0.0370	0.0272	0.0380	0.134	0.235
Total time	0.463	14.916	0.681	2.989	6.311	30.692

Table 10a. M2.4 (tolerance = 10^{-10})

Craig's Method: $\frac{\lambda_i}{\mu_i} = 1 - \beta_i h_i^{\alpha}$, $\mu_i = s_i = 1$, $\beta_i = \frac{1}{8}$, $i=1,2$, $\alpha = 2$												
(n_1,n_2)	β											
	1.00	.75	.50	.25	.10	.01	.001	0	A_0	-.01	-.25	-.75
(8,8)	8	8	8	8	8	8	9	10	7	8	9	9
(16,16)	16	16	15	13	12	9	9	12	8	9	16	22
(32,32)	28	26	23	20	15	10	10	14	10	10	23	39
(64,64)	48	43	37	28	20	12	11	17	12	12	38	**

** more than 50 iterations

Table 10b. M2.4 (tolerance = 10^{-10})

Orthodir Method: $\frac{\lambda_i}{\mu_i} = 1 - \beta_i h_i^{\alpha}$, $\mu_i = s_i = 1$, $\beta_i = \frac{1}{8}$, $i=1,2$, $\alpha = 2$												
(n_1,n_2)	β											
	1.00	.75	.50	.25	.10	.01	.001	0	A_0	-.01	-.25	-.75
(8,8)	7	8	8	8	8	8	**	9	7	8	8	8
(16,16)	15	15	15	16	16	16	**	17	15	16	16	16
(32,32)	24	23	22	21	21	21	**	26	21	21	23	28
(64,64)	36	33	31	27	25	24	**	30	25	25	32	44

** more than 50 iterations

Table 11a. M2.3.2 (tolerance = 10^{-10})

Orthodir Method: $s_l\,\mu_l = \lambda_l + \eta_l\,h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $i=1,2$, $\alpha = 1$											
(n_1,n_2)	s_l			(n_1,n_2)	s_l			(n_1,n_2)	s_l		
	1	4	n_l-1		1	4	n_l-1		1	4	n_l-1
(5,8)	11	11	10	(8,8)	7	7	7	(8,5)	11	11	10
(10,16)	18	18	15	(16,16)	15	15	13	(16,10)	18	18	15
(20,32)	23	23	17	(32,32)	22	22	••	(32,20)	23	23	17
(40,64)	26	26	••	(64,64)	26	27	••	(64,40)	26	26	••

•• more than 50 iterations

Table 11b. M2.3.2 (tolerance = 10^{-10})

Orthodir Method: $s_l\,\mu_l = \lambda_l + \eta_l\,h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $i=1,2$, $\alpha = 2$											
(n_1,n_2)	s_l			(n_1,n_2)	s_l			(n_1,n_2)	s_l		
	1	4	n_l-1		1	4	n_l-1		1	4	n_l-1
(5,8)	11	11	10	(8,8)	7	7	7	(8,5)	11	11	10
(10,16)	18	17	14	(16,16)	15	15	13	(16,10)	18	17	14
(20,32)	23	23	17	(32,32)	21	21	••	(32,20)	23	23	17
(40,64)	26	26	••	(64,64)	26	26	••	(64,40)	26	26	••

•• more than 50 iterations

Table 12. M2.5.1 (tolerance = 10^{-6})

Orthodir Method: $s_l\,\mu_l = \lambda_l + \eta_l\,h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $i=1,2,3$													
(n_1,n_2,n_3)	N	s_l	α			s_l	α			s_l	α		
			1.	2.	3.		1.	2.	3.		1.	2.	3.
(4,4,4)	64	1	10	10	10	3	9	9	9	3	9	9	9
(8,8,8)	512	1	14	14	14	3	14	14	14	6	13	13	13
(16,16,16)	4096	1	18	18	18	3	18	18	18	9	17	17	17

Table 13. M2.5.1 (tolerance = 10^{-6}, time in seconds)

Orthodir Method: $s_l\,\mu_l = \lambda_l + \eta_l\,h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $s_l = 3$, $i=1,2,3$, $\alpha = 2$			
n_l	4	8	16
Initialization	0.015	0.065	0.308
Iteration	0.362	3.492	28.407
No. of iteration	9	14	18
Time per iteration	0.0402	0.249	1.578
Generating p	0.039	0.407	4.526
Total time	0.416	3.964	33.241

Table 14. M2.5.1 (tolerance = 10^{-6})

$s_l\ \mu_l = \lambda_l + \eta_l\ h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, i=1,2,3, $\alpha = 2$											
Method			point SOR: Initial guess $p_{ij} = 0$								Orthodir
n_l	s_l	N	ω^*	Relaxation factor ω							Iterations
				1.0	1.4	1.5	1.6	1.7	1.8	1.9	ω^*	
4	1	64	1.700	474	206	159	115	69	190	••	69	10
4	3	64	1.593	183	76	54	34	53	104	••	30	9
8	1	512	1.831	••	••	907	660	489	300	••	242	14
8	7	512	1.715	458	199	154	111	64	77	305	49	12

•• more than 1000 iterations.

Table 15. M2.5.1 (tolerance = 10^{-6}, time in seconds)

Parameter	$s_l\ \mu_l = \lambda_l + \eta_l\ h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, i=1,2,3, $\alpha = 2$					
Problem	$n_l = 4$, $s_l = 3$		$n_l = 8$, $s_l = 7$		$n_l = 8$, $s_l = 1$	
Dimension N	64		512		512	
Method	Orthodir	pt SOR	Orthodir	pt SOR	Orthodir	pt SOR
ω^*	...	1.593	...	1.715	...	1.831
No. of iterations	9	30	12	49	14	242
Time for iteration	0.364	0.498	2.822	5.936	3.365	31.225
Time per iteration	0.0405	0.0166	0.2352	0.1211	0.2404	0.1290
Total time	0.420	0.529	3.274	5.997	3.815	31.282

Table 16a. M2.6.1 (tolerance = 10^{-6})

Orthodir Method: $s_l\,\mu_l = \lambda_l + \eta_l\,h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $i=1,2$						
(n_1,n_2)	$\alpha = 1.$		$\alpha = 2.$		$\alpha = 3.$	
	(s_1,s_2)	Iterations	(s_1,s_2)	Iterations	(s_1,s_2)	Iterations
(10,10)	(2,2)	11	(5,5)	10	(4,4)	11
(20,20)	(4,4)	15	(5,5)	15	(8,8)	13
(40,40)	(8,8)	17	(5,5)	18	(16,16)	16
(80,80)	(16,16)	20	(5,5)	22	(32,32)	18

Table 16b. M2.6.1 (tolerance = 10^{-6})

Orthodir Method: $s_l\,\mu_l = \lambda_l + \eta_l\,h_l^{\alpha}$, $\lambda_l = \eta_l = 1$, $i=1,2$						
(s_1,s_2)	$\alpha = 3.$		$\alpha = 2.$		$\alpha = 2.$	
	(n_1,n_2)	Iterations	(n_1,n_2)	Iterations	(n_1,n_2)	Iterations
(10,10)	(17,18)	12	(27,28)	15	(15,15)	11
(10,10)	(25,26)	14	(35,36)	16	(30,30)	16
(10,10)	(41,42)	17	(51,52)	18	(60,60)	19
(10,10)	(73,74)	20	(83,84)	21	(120,120)	23

Appendix

A.1 Alternatives (B) & (C)

For the models discussed in § 2.3.1, there are two other alternatives to compute the quantity $E\,A_0^+\,E\,y$. They both have advantages over Alternative A. The choice of the best alternative depends on the problem itself.

Alternative (B):- Partial diagonalization of A_0 by $(S_1\,Q_1 \otimes I)$

Recall that by (2.1.5) and (2.1.12), we have

$$T = (Q_1^*\,S_1^{-1} \otimes I)\,A_0\,(S_1\,Q_1 \otimes I) = (\Gamma_1 \otimes I + I \otimes G_2).$$

Here $T = \mathrm{diag}(T_1, \ldots, T_{n_1})$, and each block $T_j = \gamma_{1,j}\,I_{n_2} + G_2$ is a tridiagonal matrix. Since $\gamma_{1,n_1} = 0$, $T_{n_1} = G_2$ is singular. Since A_0 has only a one dimensional null-space, all the other T_j, $1 \le j < n_1$, are non-singular. By (2.1.12) and (2.1.15), the generalized inverse of T_{n_1} is given by

$$T_{n_1}^+ = G_2^+ = S_2\,Q_2\,\Gamma_2^+\,Q_2^*\,S_2^{-1}. \tag{A.1.1}$$

Since each T_j, $1 \le j < n_1$, is nonsingular, the generalized inverse of T is given by

$$T^+ = \mathrm{diag}\,(T_1^{-1}, \ldots, T_{n_1-1}^{-1}, S_2\,Q_2\,\Gamma_2^+\,Q_2^*\,S_2^{-1}).$$

Thus the generalized inverse of A_0 is given by

$$A_0^+ = (S_1\,Q_1 \otimes I)\,T^+\,(Q_1^*\,S_1^{-1} \otimes I).$$

By (2.3.11), a straightforward computation shows that for any vector $y \in R^{n_2}$,

$$E^* A_0^+ E y = \sum_{j=1}^{n_1-1} ({}^1q_{n_1,j})^2 T_j^{-1} y + ({}^1q_{n_1,n_1})^2 S_2 Q_2 \Gamma_2^+ Q_2^* S_2^{-1} y. \tag{A.1.2}$$

Since for each $1 \le j < n_2$, T_j is tridiagonal, computing $z_j = T_j^{-1} y$ requires $5n_2$ operations. Thus the first term can be evaluated in $6(n_1-1)n_2$ operations, provided that $({}^1q_{n_1,j})^2$ are computed and stored before the iteration. We claim that the second term can be computed in $O(n_2)$ rather than $O(n_2^2)$ operations. We first prove

LEMMA A.1.1

$R^{n_2} = \{S_2^2 1_2\} \oplus Im(G_2)$, and the projection Py of any $y \in R^{n_2}$ onto $Im(G_2)$ is given by

$$Py = y - (1_2^* y) S_2^2 1_2. \tag{A.1.3}$$

Proof: The first statement follows directly from the spectral decomposition of G_2 in (2.1.12). To prove (A.1.3), we first write $Q_2^* S_2^{-1} y = \begin{bmatrix} y_1 \\ \beta \end{bmatrix}$, where β is a scalar. By (2.1.12) and the fact that $\gamma_{2,n_2} = 0$, $S_2 Q_2 \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$ is the null-vector of G_2, where $\mathbf{0}$ is the zero vector of length $n_2 - 1$. Thus by (2.1.10),

$$S_2 Q_2 \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = S_2^2 1_2. \tag{A.1.4}$$

This gives $[\mathbf{0}^*, 1] = 1_2^* S_2 Q_2$. Thus $\beta = [\mathbf{0}^*, 1] \cdot \begin{bmatrix} y_1 \\ \beta \end{bmatrix} = 1_2^* S_2 Q_2 Q_2^* S_2^{-1} y = 1_2^* y$. Hence

$$P y = S_2 Q_2 \begin{bmatrix} y_1 \\ 0 \end{bmatrix} = S_2 Q_2 \begin{bmatrix} y_1 \\ \beta \end{bmatrix} - \beta S_2 Q_2 \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = y - (1_2^* y) S_2^2 1_2. \quad \square$$

We note that Py can be computed in $2n_2$ operations if $S_2^2 1_2$ is stored beforehand. By (A.1.3), (A.1.4) and (2.1.15),

the second term in (A.1.2) becomes

$$S_2\, Q_2\, \Gamma_2^+\, Q_2^*\, S_2^{-1}\, y = S_2\, Q_2\, \Gamma_2^+\, Q_2^*\, S_2^{-1}\, (P\, y + (1_2^*\, y) S_2^2\, 1_2)$$
$$= S_2\, Q_2\, \Gamma_2^+\, Q_2^*\, S_2^{-1}\, P\, y + \gamma\, (1_2^*\, y)\, S_2^2\, 1_2, \qquad \text{(A.1.5)}$$

where γ is the arbitrary fixed constant given in (2.1.15). The second term in (A.1.5) can be obtained in n_2 operations if $\gamma \neq 0$. Notice that, by the definition of P, the first term $z = S_2\, Q_2\, \Gamma_2^+\, Q_2^*\, S_2^{-1}\, P\, y$ is the unique element in $Im(G_2)$ such that $G_2\, z = P\, y$. Moreover by (2.1.10), for all scalar η, $G_2\, (z + \eta\, S_2^2\, 1_2) = P\, y$. Since $S_2^2\, 1_2$ is a positive vector, there exists an η_0 such that the first entry of $\hat{z} = (z + \eta_0\, S_2^2\, 1_2)$ is 1. Using the fact that G_2 is tridiagonal, the remaining entries of $\hat{z}$ can be solved by forward substitution. This requires $3n_2$ operations. Since $z \in Im(G_2)$ is the projection of $\hat{z}$ onto $\mathrm{Im}(G_2)$, by lemma A.1.1, $z = \hat{z} - (1_2^*\, \hat{z})\, S_2^2\, 1_2$. This can be computed in another $2n_2$ operations. Thus the second term in (A.1.2) can be computed in at most $8n_2$ operations.

Hence we see that $E^*\, A_0^+\, E\, y$ can be computed in about $6n_1n_2$ operations. Using the fact that T_j has constant subdiagonal entries, i.e., $(T_j)_{l,l-1} = -\lambda_2$, we can reduce the operations count to $4n_1n_2$. In fact, instead of solving $T_j^{-1}\, y$ in (A.1.2), we can solve $w_j = (\frac{1}{\lambda_2}\, T_j^{-1})\, y$. The subdiagonal entries of this matrix are all -1, so that w_j can be computed in $3n_2$ operations rather than the usual $5n_2$ operations. Notice that (A.1.2) becomes

$$E^*\, A_0^+\, E\, y = \sum_{j=1}^{n_1-1} (\lambda_2\, {}^1q_{n_1,j}^2)\, w_j + ({}^1q_{n_1,n_1})^2\, [z + \gamma\, (1_2^*\, y) S_2^2\, 1_2].$$

With $(\lambda_2\, {}^1q_{n_1,j}^2)$ computed and stored beforehand, this computations requires only $4n_1n_2$ operations. If $n_1 << n_2$, this alternative requires much less work compared to Alternative A. Notice also that it is not necessary to generate and store $\{Q_2,\Gamma_2\}$. Recall that in alternative A, we need to store both Q_1 and Q_2. Thus this alternative saves half the storage spaces compared to Alternative A. Moreover, when $s_1 = 1$, only $O(n_1)$ storage spaces are required.

The following alternative, on the other hand, does not require the eigenpair $\{Q_1\, ,\, \Gamma_1\}$.

Alternative (C):- Partial diagonalization of A_0 by $(I \otimes S_2\, Q_2)$

Recall that by (2.1.5) and (2.1.12), we have

$$(I \otimes Q_2^* S_2^{-1})\, A_0\, (I \otimes S_2\, Q_2) = (G_1 \otimes I + I \otimes \Gamma_2).$$

Let P be the permutation matrix of order $N = n_1 n_2$ such that

$$P^*(U \otimes V)\, P = V \otimes U,$$

for any U of order n_1 and V of order n_2. By (2.1.5) and (2.1.12), we have

$$\Psi = P^*\, (I \otimes Q_2^* S_2^{-1})\, A_0\, (I \otimes S_2\, Q_2)\, P = (\Gamma_2 \otimes I + I \otimes G_1).$$

Here $\Psi = \text{diag}\,(\Psi_1, \ldots, \Psi_{n_2})$, and each block $\Psi_j = G_1 + \gamma_{2,j} \cdot I_{n_1}$ is a tridiagonal matrix. Since $\gamma_{2,n_2} = 0$, $\Psi_{n_2} = G_1$ is singular. All the other Ψ_j, $1 \le j < n_2$, are nonsingular as A_0 has only one dimensional null-space. Similar to (A.1.1), we have

$$G_1^+ = \Psi_{n_2}^+ = S_1\, Q_1\, \Gamma_1^+\, Q_1^*\, S_1^{-1},$$

where

$$\Gamma_1^+ = \text{diag}\,(\gamma_{1,1}^{-1}, \ldots, \gamma_{1,n_1-1}^{-1}, \gamma),$$

with γ defined arbitrarily. The generalized inverses of Ψ and A_0 are

$$\Psi^+ = \text{diag}\,(\Psi_1^{-1}, \ldots, \Psi_{n_2-1}^{-1}, \Psi_{n_2}^+), \tag{A.1.6}$$

and

$$A_0^+ = (I \otimes S_2\, Q_2)\, P\, \Psi^+\, P^*\, (I \otimes Q_2^* S_2^{-1}) \tag{A.1.7}$$

respectively. Hence

$$\begin{aligned} E^*\, A_0^+\, E &= E\, (I \otimes S_2\, Q_2)\, P\, \Psi^+\, P^*\, (I \otimes Q_2^* S_2^{-1})\, E \\ &= S_2\, Q_2\, E^*\, P\, \Psi^+\, P^*\, E\, Q_2^* S_2^{-1}. \end{aligned}$$

A straightforward computation gives

$$E^*\, P\, \Psi^+\, P^*\, E = \Omega = \text{diag}\,(\omega_1, \ldots, \omega_{n_2}),$$

where

$$\omega_j = {}^1e_{n_1}^*\, \Psi_j^{-1}\, {}^1e_{n_1} = {}^1e_{n_1}^*\, (G_1 + \gamma_{2,j}\, I_{n_2})^{-1}\, {}^1e_{n_1}, \qquad 1 \le j < n_2, \tag{A.1.8}$$

and

$$\omega_{n_2} = {}^1e_{n_1}^* \Psi_{n_2}^+ {}^1e_{n_1}.$$

Thus

$$E^* A_0^+ E = S_2 Q_2 \Omega Q_2^* S_2^{-1}.$$

Compare this with (2.3.15), we see that $\Omega = \Phi$. By lemma 2.3.2, ω_{n_2} can also be defined arbitrarily.

Thus, before we start our iteration, we generate and store $\{Q_2,\Gamma_2\}$. We then solve and save ω_j according to (A.1.8). Since $(G_1 + \gamma_{2j} I)$ is a tridiagonal matrix, and only the last entry of the solution is required, ω_j can be generated in $3 n_1$ operations for each j. With Ω stored before the iteration, the computation of $E^* A_0^+ E y$ can be done in $2 n_2^2 + O(n_2)$ operations. We remark that there is no need to generate the eigenpair $\{Q_1,\Gamma_1\}$. Thus it saves half the storage spaces compared with Alternative A. Clearly this will be the best choice when $n_2 << n_1$. If $s_2 = 1$, this algorithm needs only $O(n_2)$ work and storage per iteration.

A.2 An Algorithm for the Model in § 2.3.2

In this section, we give an algorithm for computing the product $E^* \bar{R} A_0^+ E y$ given in (2.3.36). This algorithm requires about $6n^2$ operations and n^2 memory. In the single server case, they can be reduced to $5n^2$ and $O(n)$ respectively. The idea is to diagonalize A_0^+ partially by $(I \otimes S_2 Q_2)$ as in Appendix A.1, and accumulate the result blockwise as mentioned in § 2.3.1.

By (2.3.33), we have, for any $y \in R^m$,

$$E y = {}^1y \otimes {}^2e_{n_2} + {}^1e_{n_1} \otimes {}^2y, \tag{A.2.1}$$

where ${}^iy \in R^{n_i}$ such that $\sum_{ij} ({}^iy)_j = 0$. Here we use $(\cdot)_j$ to denote the j-th entry of the vector. Since $(Ey)_{ij} = ({}^1y)_{n_1} + ({}^2y)_{n_2}$, we can always fix one of $({}^iy)_{n_i}$ arbitrarily. Let us set $({}^1y)_{n_1} = 0$. By (A.1.7), we have

$$E^* \bar{R} A_0^+ E y = E^* \bar{R} (I \otimes S_2 Q_2) P \Psi^+ P^* (I \otimes Q_2^* S_2^{-1}) E y. \tag{A.2.2}$$

Let us divide this computation into three steps:

Step I. Computation of $\bar{u} = P^* (I \otimes Q_2^* S_2^{-1}) E y$.

We first note that by (A.2.1) and the definition of P,

$$\bar{u} = P^* (I \otimes Q_2^* S_2^{-1}) E y = Q_2^* S_2^{-1} \cdot {}^2e_{n_2} \otimes {}^1y + Q_2^* S_2^{-1} \cdot {}^2y \otimes {}^1e_{n_1}. \tag{A.2.3}$$

Thus if $Q_2^* S_2^{-1} \cdot {}^2e_{n_2}$ is generated and stored before the iteration, we only need to compute $Q_2^* S_2^{-1} \cdot {}^2y$ here. This requires only $n_2^2 + n_2$ operations. It also need n_2^2 memory spaces to hold Q_2 and two linear arrays for $Q_2^* S_2^{-1} \cdot {}^2e_{n_2}$ and $Q_2^* S_2^{-1} \cdot {}^2y$. When $s_2 = 1$, this step requires $n_2 \log n_2 + n_2$ operations and $O(n_2)$ storage.

Step II. Computation of $\bar{v} = \Psi^* \bar{u}$.

Let us partition $\bar{u}$ in (A.2.3) as $\bar{u} = [\, u_1, \ldots, u_{n_2} \,]$, where each block u_l is of length n_1. Notice that there is no need to store the whole vector $\bar{u}$ which is of length $n_1 n_2$. With $Q_2^* S_2^{-1} \cdot {}^2e_{n_2}$ and $Q_2^* S_2^{-1} \cdot {}^2y$ stored, we can generate u_l one at a time by (A.2.3). Let us define $\bar{v} = \Psi^+ \bar{u}$, and partition $\bar{v}$ as in $\bar{u}$. Then by (A.1.6), we have,

$$v_j = \begin{cases} \Psi_j^{-1} u_j & 1 \le j < n_2 \\ \Psi_{n_2}^{+} u_{n_2} & j = n_2 \end{cases}. \tag{A.2.4}$$

Since Ψ_j, $1 \le j < n_2$, are tridiagonal, each v_j can be computed in $5n_1$ operations. Using the fact that the Ψ_j have constant subdiagonals, this can be reduced to $3n_1$ operations. v_{n_2} can be computed by using the method mentioned in § A.1, Alternative B, i.e., we fix the first entry of v_{n_2} and solve for the other entries by forward substitution. This requires only $8n_1$ operations. Thus the whole step can be done in approximately $3n_1 n_2$ operations.

Step III. Computation of $E^* \bar{R} (I \otimes S_2 Q_2) P \bar{v}$.

Let us write $\bar{w} = [\, w_1, \ldots, w_{n_1} \,] = P \bar{v}$, where each block w_l is a vector of length n_2. Notice that $(w_l)_j = (v_j)_l$. Thus no computation is required for w. Notice also that

$$(I \otimes S_2 Q_2)\bar{w} = [\, S_2 Q_2 w_1, \ldots, S_2 Q_2 w_{n_1}].$$

Since this product will be multiplied by $E^* \bar{R}$, we only need to compute the projection of $(I \otimes S_2 Q_2)\, \bar{w}$ onto $\mathrm{Im}(\bar{R})$. In fact, if we let

$$t = [\, (S_2 Q_2 w_1)_{n_2}, \ldots, (S_2 Q_2 w_{n_1-1})_{n_2}, 0 \,] \in R^{n_1},$$

we then have

$$E^* \bar{R} (I \otimes S_2 Q_2)\bar{w} = E^* \bar{R} [\, t \otimes {}^2e_{n_2} + {}^1e_{n_1} \otimes S_2 Q_2 w_{n_1}]. \tag{A.2.5}$$

Thus we only have to compute $(S_2 Q_2 w_l)_{n_2}$, $1 \le i < n_1$, and $S_2 Q_2 w_{n_1}$. Notice that

$$(S_2 Q_2 w_l)_{n_2} = \sum_{j=1}^{n_2} ({}^2q_{n_2 j}\, {}^2d_j)(w_l)_j = \sum_{j=1}^{n_2} ({}^2q_{n_2 j}\, {}^2d_j)\, (v_j)_l, \quad 1 \le i < n_1, \tag{A.2.6}$$

and

$$(S_2\, Q_2\, w_{n_1})_i = \sum_{j=1}^{n_2} ({}^2q_{i,j}\ {}^2d_j)(w_{n_1})_j = \sum_{j=1}^{n_2} ({}^2q_{i,j}\ {}^2d_j)\ (v_j)_{n_1},\quad 1 \le i < n_2. \tag{A.2.7}$$

Thus it is clear that if $({}^2q_{n_2,j}\ {}^2d_j)$ is computed and stored before the iteration, this step requires only $2n_2^2$ operations. Moreover, there is no need to generate and store the whole vector $\bar{v}$ which is of length n_1n_2, in step II before we start computing t. In fact, we can generate the v_j by (A.2.4) and accumulate their contributions to t and $(S_2\, Q_2 w_{n_1})$, one at a time, according to (A.2.6) and (A.2.7).

Notice that since $\bar{R}$ is sparse, the matrix-vector product of $\bar{R}$ with the vector in (A.2.5) can be computed in $O(n_l)$ operations. Thus combining steps I, II and III, we conclude that this algorithm requires only $3n_2^2 + 3n_1n_2$ operations and n_2^2 storage spaces for Q_2 and some storage spaces for vectors of length n_1 or n_2. When $s_2 = 1$, there is no need to hold Q_2, so the memory requirement is then linear in n_l. When only $s_1 = 1$, we notice that this problem is symmetric with respect to the queues. We can repeat the argument, diagonalizing A_0^+ partially by $(I \otimes Q_1^*\, S_1^{-1})$ rather than by $(I \otimes Q_2^*\, S_2^{-1})$. This will give us the same count.

A.3 An Algorithm for the Model in § 2.5.1

We will give an algorithm that computes $R_0 A_0^+ \xi$ in (2.5.6) in $O(n_l^3)$ work and $O(n_l^2)$ storage spaces. The idea is exactly the same as in § A.2., i.e., we diagonalize A_0^+ partially and accumulate the results blockwise. To begin with, let us define the permutation matrix P as the matrix of dimension $N = n_1 n_2 n_3$, such that

$$P (U \otimes V \otimes W) P^* = V \otimes W \otimes U,$$

for any U, V and W of order n_1, n_2 and n_3 respectively. Moreover, let

$$T = (I \otimes I \otimes S_3 Q_3) (I \otimes S_2 Q_2 \otimes I) P = (I \otimes S_2 Q_2 \otimes S_3 Q_3) P.$$

Using (2.1.22), we have,

$$\Psi = T^{-1} A_0 T = (I \otimes \Gamma_3 \otimes I) + (\Gamma_2 \otimes I \otimes I) + (I \otimes I \otimes G_1).$$

Here

$$\Psi = \text{diag}\,(\Psi_{1,1}, \ldots, \Psi_{1,n_3}, \Psi_{2,1}, \ldots, \Psi_{n_2,n_3}), \tag{A.3.1}$$

with tridiagonal matrices

$$\Psi_{j,k} = \gamma_{2,j} I + \gamma_{3,k} I + G_1. \tag{A.3.2}$$

We remark that only $\Psi_{n_2,n_3} = G_1$ is singular since A_0 has only a one dimensional null-space. Similar to § A.2, we define the generalized inverse of Ψ as

$$\Psi^+ = (\Psi_{1,1}^{-1}, \ldots, \Psi_{n_2,n_3-1}^{-1}, \Psi_{n_2,n_3}^{+}), \tag{A.3.3}$$

where

$$\Psi_{n_2,n_3}^{+} = S_1 Q_1 \Gamma_1^+ Q_1^* S_1^{-1}. \tag{A.3.4}$$

Therefore

$$R_0 A_0^+ \xi = R_0 T \Psi^+ T^{-1} \xi.$$

We divide the calculation of the cost into three steps. The first step calculates the cost of computing $t = T^{-1} \xi$, the second step the cost of computing $u = \Psi^+ t$ and the final step the cost of computing $R_0 T u$.

STEP I. Computation of $t = T^{-1}\xi$.

For simplicity, we use ${}^{j}x$ to denote a vector of length $n_3 n_j$ partitioned into n_j blocks. More precisely,

$$ {}^{j}x = [{}^{j}x_1, \ldots, {}^{j}x_{n_j}]^* $$

where each ${}^{j}x_k$, $1 \le k \le n_j$ is a n_3-vectors. We also define for any n_2-vector e and any $n_1 n_3$-vector ${}^{1}x$, the product $\oplus$ as,

$$ e \oplus {}^{1}x = [e \otimes {}^{1}x_1, \ldots, e \otimes {}^{1}x_{n_1}]^*. $$

This is an N-vector, with each $e \otimes {}^{1}x_k$ an $n_2 n_3$-vector. By (2.5.3), each $x \in Im(R_0)$ can be written as

$$ x = {}^{2}e_{n_2} \oplus {}^{1}x + {}^{1}e_{n_1} \otimes {}^{2}x, $$

where ${}^{j}e_k$ are the k-th unit vector in R^{n_j}. Since the last block of x is the sum of ${}^{1}x_{n_1}$ and ${}^{2}x_{n_2}$, one of these vectors can be set to zero. Without loss of generality, let us set ${}^{1}x_{n_1} = 0$ for all $x \in Im(R_0)$. In particular, since $\xi \in Im(R_0)$,

$$ \xi = {}^{2}e_{n_2} \oplus {}^{1}\xi + {}^{1}e_{n_1} \otimes {}^{2}\xi, $$

where ${}^{1}\xi_{n_1} = 0$. Notice that

$$ y = (I \otimes I \otimes Q_3^* S_3^{-1})\, \xi = {}^{2}e_{n_2} \oplus {}^{1}y + {}^{1}e_{n_1} \otimes {}^{2}y, $$

where

$$ {}^{j}y_k = Q_3^* S_3^{-1} ({}^{j}\xi_k), \qquad 1 \le k \le n_j, \ j = 1,2. $$

This computation requires $(n_1 + n_2)n_3^2$ operations. Next we observe that

$$ z = (I \otimes Q_2^* S_2^{-1} \otimes I)\, y = (Q_2^* S_2^{-1}\, {}^{2}e_{n_2}) \oplus {}^{1}y + {}^{1}e_{n_1} \otimes {}^{2}z, \qquad \text{(A.3.5)} $$

where

$$ {}^{2}z = (Q_2^* S_2^{-1} \otimes I)\, ({}^{2}y). $$

Therefore

$$ {}^{2}z_j = \sum_{k=1}^{n_2} {}^{2}q_{kj} ({}^{2}d_k)^{-1} ({}^{2}y_k) \qquad 1 \le j \le n_2, $$

where ${}^2q_{k,j}$ are the (k,j) entry of Q_2, and ${}^2d_k$ are given in (2.1.7). Thus the second term in (A.3.5) can be computed in $n_2^2 n_3$ operations. If $Q_2^* S_2^{-1}\, {}^2e_{n_2}$ is computed and stored before we iterate, the first term in (A.3.5) requires N operations. Thus z can be generated in $N + n_2^2 n_3$ operations.

Next, let $t = P^* z$. This can be done by indexing and requires no extra cost. In fact, if we partition t into $n_2 n_3$ blocks according to the partition of Ψ in (A.3.1), and z into $n_1 n_2$ blocks as $z = [z_{1,1}, \ldots, z_{1,n_2}, \ldots, z_{n_1,n_2}]^*$, where each $z_{k,j}$ is an n_3-vector; then

$$(t_{j,l})_k = (z_{k,j})_l, \qquad 1 \le k \le n_1,\ 1 \le j \le n_2,\ 1 \le l \le n_3. \tag{A.3.6}$$

Here we have used $(\cdot)_j$ to denote the j-th entry. Thus the cost for computing $t = T^{-1}\xi$ is

$$(n_1 + n_2) n_3^2 + N + n_2^2 n_3 \quad \text{operations.} \tag{A.3.7}$$

STEP II. Computation of $u = \Psi^+ t$.

Next let us calculate the cost of computing $u = \Psi^+ t = \Psi^+ T^{-1} \xi$. From (A.3.1) and (A.3.2), we have to solve for $u_{j,k}$ in

$$\Psi_{j,k}\, u_{j,k} = (\gamma_{2,j} I + \gamma_{3,k} I + G_1)\, u_{j,k} = t_{j,k} \qquad 1 \le j \le n_2,\ 1 \le k \le n_3. \tag{A.3.8}$$

If $j \neq n_2$ or $k \neq n_3$, $\Psi_{j,k}$ is tridiagonal and invertible, hence each $u_{j,k}$ can be generated within $5n_1$ operations. Using the fact that G_1 has constant sub-diagonal, this can be reduced to $3n_1$ operations, see Appendix A.1 for more detail. If $j = n_2$ and $k = n_3$, by (A.3.4), we have to solve

$$S_1\, Q_1\, \Gamma_1^+\, Q_1^*\, S_1^{-1}\, u_{n_2,n_3} = t_{n_2,n_3}, \tag{A.3.9}$$

which requires only $2n_1^2$ operations. If we use the method mentioned in Appendix A.1, this count will be reduced to $O(n_1)$ operations. Thus the total cost of generating $u_{j,k}$ for all j,k is approximately

$$3\,N \quad \text{operations.} \tag{A.3.10}$$

STEP III. Computation of $R_0\, T\, u$.

Notice that $R_0\, T\, u = R_0\,(I \otimes I \otimes S_3\, Q_3)\,(I \otimes S_2\, Q_2 \otimes I)\, P\, u$. Let $v = P\, u$. If we partition v as $v = [v_{1,1}, \ldots, v_{1,n_2}, \ldots, v_{n_1,n_2}]^*$ where $v_{k,j}$ are n_3-vector, then

$$(v_{k,j})_l = (u_{j,l})_k \qquad 1 \le k \le n_1,\ 1 \le j \le n_2,\ 1 \le l \le n_3. \tag{A.3.11}$$

Thus this step can be done without any arithmetic cost. Next we observe that since R_0 is sparse, it suffices to consider the cost of computing $E\, T\, u$, where E is the projection operator onto $Im(R_0)$. By (2.5.3), this projection operator satisfies

$$E\,(I \otimes I \otimes S_3\, Q_3)\,(I \otimes S_2\, Q_2 \otimes I) = (I \otimes I \otimes S_3\, Q_3)\, E\,(I \otimes S_2\, Q_2 \otimes I).$$

Let $w = E\,(I \otimes S_2\, Q_2 \otimes I)\, v$. Since $E\, T\, u \in Im(R_0)$, we have

$$w = {}^2e_{n_2} \oplus {}^1w + {}^1e_{n_1} \otimes {}^2w, \tag{A.3.12}$$

where

$${}^1w_j = \begin{cases} ({}^2d_{n_2}) \sum_{k=1}^{n_2} {}^2q_{n_2,k}\, v_{j,k} & 1 \le j < n_1, \\ 0 & j = n_1, \end{cases} \tag{A.3.13}$$

and

$${}^2w_j = ({}^2d_j) \cdot \sum_{k=1}^{n_2} {}^2q_{j,k}\, v_{n_1,k}, \qquad 1 \le j \le n_2. \tag{A.3.14}$$

Clearly the computation of ${}^1w_j$ requires N operations while ${}^2w_j$ requires $n_2^2\, n_3$ operations. Finally let

$$r = E\, A_0^+\, \xi = (I \otimes I \otimes S_3\, Q_3)\, w = {}^2e_{n_2} \oplus {}^1r + {}^1e_{n_1} \otimes {}^2r,$$

where

$${}^jr_k = S_3\, Q_3\, {}^jw_k, \qquad 1 \le k \le n_j,\ j = 1,2.$$

This step requires $(n_1 + n_2)n_3^2$ operations. Thus $E\, T\, u$ can be generated in

$$N + n_2^2 n_3 + (n_1 + n_2)n_3^2 \text{ operations.} \tag{A.3.15}$$

Combining (A.3.7), (A.3.10) and (A.3.15), we conclude that the product $E\,A_0^+\,\xi$ can be generated in

$$5\,N + 2\,n_2^2\,n_3 + 2(n_1 + n_2)\,n_3^2 \text{ operations.} \tag{A.3.16}$$

When $n_1 = n_2 = n_3$, this count is $11N$. Since R_0 is sparse, the dominant work in each iteration is the computation of $E\,A_0^+\,\xi$. Thus we see that the work per iteration is approximately $11N$. We remark that this can be further reduced to $10N$. In fact, we can combine the cross product in the first term in (A.3.5) and the cross product in the first term in (A.3.12).

We note that it is not necessary to generate and store all the entries of t and u before we can compute $\Psi^+\,t$ and $T\,u$ respectively. In fact, we can first generate $t_{k,l}$, one at a time, by (A.3.5) and (A.3.6). In (A.3.5), we only need to store ${}^1y$ and ${}^2z$, which requires only $(n_1 + n_2 - 1)n_3$ storage. After $t_{k,l}$ is generate, we can compute $u_{k,l}$ by (A.3.8) or (A.3.9), and accumulate the result of $T\,u$ to the corresponding locations before we generate another $t_{k,l}$. More precisely, by (A.3.11), (A.3.13) and (A.3.14), right after $u_{k,l}$ is generated, we can update the l-entry of every ${}^1w_j$ and ${}^2w_j$ in the following manner:

$$({}^1w_j)_l = ({}^2d_{n_2}) \sum_{k=1}^{n_2} {}^2q_{n_2,k}\,(v_{j,k})_l = ({}^2d_{n_2}) \sum_{k=1}^{n_2} {}^2q_{n_2,k}\,(u_{k,l})_j, \qquad 1 \le j < n_1,\ 1 \le l \le n_3,$$

and

$$({}^2w_j)_l = ({}^2d_j) \sum_{k=1}^{n_2} {}^2q_{j,k}\,(v_{n_1,k})_l = ({}^2d_j) \sum_{k=1}^{n_2} {}^2q_{j,k}\,(u_{k,l})_{n_1}, \qquad 1 \le j \le n_1,\ 1 \le l \le n_3.$$

It is clear that this way of accumulating the results will not increase the cost of computation, and we only need to store a few vectors of length $O(n_i^2)$ besides the Q_i's.

A.4 Fast Fourier Transform for Arbitrary n

In § 2.3.1, we have to compute

$$y_k = \sum_{j=1}^{n} z_j \, e^{\frac{ikj\pi}{n}}, \qquad 0 \le k < n. \tag{A.4.1}$$

In this section, we will show that the vector $y = [\, y_1, \ldots, y_n \,]$, can be obtained in $O(n\log n)$ operations.

We begin by rewriting (A.4.1) in a standard form. Thus let $z_j = 0$ for $n < j \le 2n$, we have

$$y_k = \sum_{j=1}^{N} z_j \, e^{\frac{2\pi ikj}{N}}, \qquad 0 \le k < n, \tag{A.4.2}$$

where $N = 2n$. Notice that if we set $\tilde{z}_j = z_{(N-j)}$, $0 \le j \le N-1$, then

$$y_k = \sum_{j=0}^{N-1} \tilde{z}_j \, e^{-\frac{2\pi ikj}{N}}, \qquad 0 \le k < n, \tag{A.4.3}$$

Thus without loss of generality, we can assume that we are using formula (A.4.3).

If n, and hence N, is a power of 2, then (A.4.3) can be computed by the regular Fast Fourier Transform. The work required is $N \log N = 2n \log n + O(n)$, see Cooley and Tukey [12]. For other values of n, we need the following standard results in Fourier analysis. The proof can be found in Brigham [9].

LEMMA A.4.1

For any N-vectors $y = [\, y_1, \ldots, y_N \,]$ and $z = [\, z_1, \ldots, z_N \,]$, define

$$(F_N z)_k = \hat{z}_k = \frac{1}{N} \sum_{j=0}^{N-1} z_j \, e^{-\frac{2\pi ikj}{N}}, \tag{A.4.4}$$

$$(z * y)_k = \sum_{j=0}^{N-1} z_j \, y_{(k-j) \bmod N}, \tag{A.4.5}$$

$$(z \cdot y)_k = z_k \, y_k, \tag{A.4.6}$$

for $0 \le k < N$. We have,

$$(1)\quad z_j = \sum_{k=0}^{N-1} \hat{z}_k \, e^{\frac{2\pi ikj}{N}}, \qquad 0 \le j < N. \tag{A.4.7}$$

Thus $\sqrt{N}\, F_N$ is a unitary matrix,

$$(2)\quad F_N (z \cdot y) = \hat{z} * \hat{y}, \tag{A.4.8}$$

$$(3)\quad F_N (z * y) = N \hat{z} \cdot \hat{y}. \quad \square \tag{A.4.9}$$

By (A.4.3), for $0 \le k < N$, we have

$$y_k = \sum_{j=0}^{N-1} \hat{z}_j \, e^{\frac{\pi i}{N}((k-j)^2 - k^2 - j^2)} = e^{-\frac{i\pi k^2}{N}} \sum_{j=0}^{N-1} \hat{z}_j \, e^{-\frac{\pi ij^2}{N}} e^{\frac{\pi i(k-j)^2}{N}} = x_k \, e^{-\frac{i\pi k^2}{N}}, \tag{A.4.10}$$

where

$$x_k = \sum_{j=0}^{N-1} \hat{z}_j \, e^{-\frac{\pi ij^2}{N}} e^{\frac{\pi i(k-j)^2}{N}}. \tag{A.4.11}$$

Notice that the vector y can be obtained in $O(n)$ operations once the vector x is computed. To compute $\{ x_k \}_{k=1}^{N}$, we first choose an $M \ge 2N-1$ such that M is a power of 2. We then define

$$\alpha_j = \begin{cases} \hat{z}_j \, e^{-\frac{\pi ij^2}{N}} & 0 \le j \le N-1, \\ 0 & \text{otherwise}, \end{cases} \tag{A.4.12}$$

and

$$\beta_j = \begin{cases} e^{\frac{\pi ij^2}{N}} & 0 \le j \le N-1, \\ \beta_{N-l} & j = M - l,\ 0 < l < N, \\ 0 & \text{otherwise}. \end{cases} \tag{A.4.13}$$

We note that by (A.4.12), (A.4.11) becomes

$$x_k = \sum_{j=0}^{M-1} \alpha_j \cdot \beta_{(k-j) \bmod N}, \quad 0 \le k < N. \tag{A.4.14}$$

Let $l = k-j$. Since $0 \le k, j < N$, $-N < l < N$. If $0 \le l < N$, then $l < M$, hence

$$\beta_{l \bmod N} = \beta_l = \beta_{l \bmod M}, \qquad \text{for } 0 \le l < N. \tag{A.4.15}$$

If $0 < -l < N$, then by (A.4.13), $\beta_{M-(-l)} = \beta_{N-(-l)}$. Hence,

$$\beta_{l \bmod M} = \beta_{M+l} = \beta_{M-(-l)} = \beta_{N-(-l)} = \beta_{l \bmod N}, \qquad \text{for } -N < l < 0. \tag{A.4.16}$$

Combining (A.4.15) and (A.4.16), (A.4.14) gives

$$x_k = \sum_{j=0}^{M-1} \alpha_j \cdot \beta_{(k-j) \bmod M} = (\alpha * \beta)_k,$$

where $\alpha = [\alpha_0, \ldots, \alpha_{M-1}]$ and $\beta = [\beta_0, \ldots, \beta_{M-1}]$. Hence by lemma A.4.1,

$$x = F_M^{-1} (M \hat{\alpha} \cdot \hat{\beta}) = F_M^{-1} (M F_M \alpha \cdot F_M \beta).$$

Thus $\{ x_k \}_{k=1}^{M}$ can be computed at the expense of two Fourier Transforms and one inverse Fourier Transform. This requires $3M \log M$ operations. Notice that for arbitrary N, there exists an M such that $2N-1 \leq M \leq 4N = 8n$ and M is a power of 2. Thus the whole computation requires $24n \log n + O(n)$ operations.

References

1. Astrakhantsev, G., *Method of Fictitious Domains for a Second-order Elliptic Equation with Natural Boundary Conditions*. U.S.S.R. Comput. Maths Math. Phys., V18, 1978, pp. 114-121.
2. Axelsson, O., *Solution of Linear Systems of Equations: Iterative Methods*. Lect. Notes in Math, #572, Springer-Verlag, New York, 1977.
3. Banegas A., *Fast Poisson Solvers for Problems with Sparsity*. Math. Comp., V32, 1978, pp. 441-446.
4. Ben-Isreal, A., & Greville, T., *Generalized Inverses: Theory and Applications*. Wiley, New York, 1974.
5. Berman, A., & Plemmons, R., *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York, 1979.
6. Bjørstad, P., *Fast Numerical Solution of the Biharmonic Dirichlet Problem on Rectangles*. SIAM J. Numer. Anal., V20, 1983, pp. 59-71.
7. Bjørstad, P., & Widlund, O., *Iterative Methods for the Solutions of Elliptic Problems on Regions Partitioned into Substructures*. NYU Comp. Sci. Dept. Tech. Rep., #136, 1984.
8. Bramble, J., Pasciak, J., & Schatz, A., *Preconditioners for Interface Problems on Mesh Domains*. To appear.
9. Brigham, E., *The Fast Fourier Transform*. Prentice-Hall, New Jersey, 1974.
10. Buzbee, B., Dorr, F., George, J., & Golub, G., *The Direct Solution of the Discrete Poisson Equations on Irregular Regions*. SIAM J. Numer. Anal., V8, 1971, pp. 722-736.
11. Concus, P., Golub, G., & O'Leary, D., *A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations*. Proc. Symposium on Sparse Matrix Computations. Bunch, J.R., & Rose, D.J., eds. Academic Press, 1976.
12. Cooley, J., & Tuckey, J., *An Algorithm for the Machine Calculation of Complex Fourier Series*. Math. Comp., V19, 1965, pp. 297-301.
13. Daniel, J., *The Conjugate Gradient Method for Linear and Nonlinear Operator Equations*. SIAM J. Numer. Anal., V4, 1967, pp. 10-26.
14. Dryja, M., *A Capacitance Matrix Method for Dirichlet Problem on Polygon Region*. Numer. Math., V39, 1982, pp. 51-64.
15. Elman, H., *Iterative Methods for Large, Sparse Nonsymmetric Systems of Linear Equations*. Ph.D thesis. Department of Computer Science, Yale University, 1982.
16. Fiedler, M. & Ptak, V., *On Matrices with Nonpositive Off-diagonal Elements and Positive Principal Minors*. Czechoslovak Math. J., V12, 1962, pp. 382-400.
17. Funderlic, R., & Mankin, J., *Solution of Homogeneous Systems of Linear Equations arising from Compartmental Models*. SIAM J. Sci. Statist. Comput., V2, 1981, pp. 375-383.
18. George & Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Englewood, NJ, 1981.
19. Grisvard, P., *Elliptic Problems in Non-smooth Regions*. Pitman, Boston, 1985.

20. Hestenes, M., & Stiefel, E., *Methods of Conjugate Gradients for Solving Linear Systems*. J. Res. Nat. Bur. Standards, V. 49, 1952, pp. 409-436.

21. Hestenes, M., *The Conjugate Gradient Method for Solving Linear Systems*. Proc of Symposia in Applied Math., VI. McGraw-Hill, 1956.

22. Kammerer, W., & Nashed, M., *On the Convergence of the Conjugate Gradient Method for Singular Linear Operator Equations*. SIAM J. Numer. Anal., V9, 1972, pp. 165-181.

23. Kaufman, L., Seery, J., & Morrison, J., *Overflow Models for Dimension PBX Feature Packages*. The Bell System Technical Journal, V60, 1981, pp. 661-676.

24. Kaufman, L., *Matrix Methods for Queueing Problems*. SIAM J. Sci. Statist. Comput., V4, 1982, pp. 525-552.

25. Kleinrock, L., *Queueing Systems. Volume 1, Theory*. Wiely, New York, 1975.

26. Lewis, J., & Rehm, R., *The Numerical Solution of a Non-separable Elliptic Partial Differential Equation by Preconditioned Conjugate Gradient*. J. Res. Nat. Bur. Standards, V85, 1980, pp. 367-390.

27. Lions, J., & Magenes, E., *Nonhomogeneous Boundary Value Problems I*. Springer Verlag, New York, 1972.

28. Luenberger, D., *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, 1973.

29. Massey, W., *Open Networks of Queues: Their Algebraic Structure and Estimating Their Transisent Behaviour*. Adv. in Appl. Probab., V16, 1984, pp. 176-201.

30. Neuts, M., *Algorithmic Methods in Probability*. North-Holland, 1977.

31. Oden, J., & Reddy, J., *An Introduction to the Mathematical Theory of Finite Elements* Wiley, New York, 1976.

32. O'Leary, D., & Widlund, O., *Capacitance Matrix Methods for the Helmholtz Equation on General Three-dimensional Regions*. Math. Comp., V33, 1979, pp. 849-879.

33. Paige, C., & Saunders, M., *Solution of Sparse Indefinite Systems of Linear Equations*. SIAM J. Num. Anal., V12, 1975, pp. 617-629.

34. Parlett, B., *The Symmetric Eigenvalue Problem*. Prentice-Hall, New Jersey, 1980.

35. Proskurowski, W., & Widlund, O., *A Finite Element-Capacitance Matrix Method for the Neumann Problem for Laplace's Equation*. SIAM J. Sci. Statist. Comput., V1, 1980, pp.410-425.

36. Rudin, W., *Real and Complex Analysis*. McGraw-Hill, New York, 1974.

37. Shieh, A., *On the Convergence of the Conjugate Gradient Method for Singular Capacitance Matrix Equations from the Neumann Problem of the Poisson Equation*. Numer. Math., V29, 1978, pp. 307-327.

38. Smith, B., et al, *Matrix Eigenvalue Routines - EISPACK Guide*. Lecture Notes in Computer Science 6, 2nd ed., Springer-Varleg, New York, 1976.

39. Van der Vorst, H., *Preconditioning by Incomplete Decomposition*. Ph. D thesis. Rijksuniversiteit te Utrecht, 1982.

40. Varga, R., *Matrix Iterative Analysis*. Prentice-Hall, New Jersey, 1962.

41. Young & Jea. *Generalized Conjugate Gradient Acceleration of Non-symmetric Iterative method*. Linear Algebra Appl., V34, 1980, pp. 159-194.

GAYLORD 142			PRINTED IN U.S.A.

NYU COMPSCI TR-171
Chan, Ryamond c.2

Iterative methods for
overflow queueing

DATE DUE	BORROWER'S NAME